CONCEPTION ET MANIPULATION DE BASES DE DONNEES DIMENSIONNELLES À CONTRAINTES

Ghozzi Faiza

N° ORDRE :

UNIVERSITE TOULOUSE III - PAUL SABATIER

U.F.R MATHEMATIQUES INFORMATIQUES GESTION

THESE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE TOULOUSE III

DISCIPLINE : ***INFORMATIQUE***

Présentée et soutenue

par

Faiza GHOZZI JEDIDI

Le 18 novembre 2004

Titre :

CONCEPTION ET MANIPULATION DE BASES DE DONNEES DIMENSIONNELLES À CONTRAINTES

Directeur de thèse : Gilles ZURFLUH

JURY

C. Chrisment	Professeur à l'Université Toulouse III,	Président
Z. Bellahsène	HDR, Maître de Conférences à l'université Montpellier II,	Rapporteur
J. M. Pinon	Professeur des Universités à l'INSA de Lyon,	Rapporteur
C. Soulé-Dupuy	Professeur à l'Université Toulouse I,	Examinateur
G. Zurfluh,	Professeur à l'Université Toulouse I,	Directeur de thèse
F. Ravat	Maître de Conférences à l'Université Toulouse I,	Co-Encadrant
O. Teste	Maître de Conférences à l'Université Toulouse III,	Co-Encadrant
R. Bouaziz	Maître Assistant à l'université de Sfax – Tunisie,	Invité

INSTITUT DE RECHERCHE EN INFORMATIQUE DE TOULOUSE

Centre National de la Recherche Scientifique (UMR 5505) - Institut National Polytechnique - Université Paul Sabatier – Université Toulouse 1

Université Paul Sabatier, 118 route de Narbonne, 31062 Toulouse cedex 04, Tel. 05.61.55.67.65

RESUME

L'accroissement du volume de données dans les systèmes d'information est de nos jours une réalité à laquelle chaque entreprise doit faire face. Notamment, elle doit permettre à ses responsables de déceler les informations pertinentes afin de prendre les bonnes décisions dans les plus brefs délais. Les systèmes décisionnels répondent à ces besoins en proposant des modèles et des techniques de manipulation des données. Dans le cadre de ces systèmes, mes travaux de thèse consistent à étudier la modélisation des données décisionnelles et à proposer un langage de manipulation adapté.

Dans un premier temps, nous proposons un modèle dimensionnel organisant les données en une **constellation** de **faits** (sujets d'analyse) associés à des **dimensions** (axes d'analyse) pouvant être partagées. Notre modèle assure une plus grande cohérence des données par sa propriété de multi instanciations qui permet de spécifier des conditions d'appartenance des instances des dimensions aux hiérarchies. De plus, nous avons défini des contraintes exprimant des relations sémantiques entre les hiérarchies intra et inter dimensions (Inclusion, Exclusion, Totalité, Partition, Simultanéité).

Au niveau de la manipulation des données, nous avons redéfini les opérateurs dimensionnels afin de permettre à l'utilisateur de mieux définir ses besoins en précisant l'ensemble des instances à analyser. Cette extension a permis d'éviter les incohérences lors de la manipulation des données dimensionnelles. Nous avons étudié également l'impact de ces contraintes sur l'optimisation des manipulations basée sur la technique de matérialisation des vues. La prise en compte des contraintes sémantiques a permis de supprimer des vues incohérentes et de réduire le nombre de vues candidates à la matérialisation.

Dans un second temps, nous proposons un processus de conception d'un schéma dimensionnel comportant une démarche descendante, basée sur les besoins des décideurs, et une démarche ascendante basée sur les données sources. Une phase de confrontation, permet d'intégrer les résultats des deux démarches pour obtenir un schéma dimensionnel en constellation intégrant à la fois les besoins des décideurs et les données sources.

Afin de valider nos propositions, nous avons développé un outil d'aide à la conception de schémas dimensionnels contraints intitulé GMAG (Générateur de MAGasin de données dimensionnelles).

Mots clés

Aide à la décision, modèle dimensionnel, contraintes, méthode de conception, algèbre d'interrogation.

À Anis & Eya

REMERCIEMENTS

***J**e tiens à remercier très sincèrement M. les Professeurs Claude Chrisment et Gilles Zurfluh, responsables de l'équipe S.I.G, pour m'avoir accueillie au sein de leur équipe afin de mener à bien cette thèse.*

Je remercie sincèrement Mme Zohra Bellahsène, Maître de conférences (HDR) à l'Université Montpellier II, pour avoir acceptée d'être rapporteur de ce mémoire, pour ses remarques pertinentes et pour l'honneur qu'elle me fait en participant au jury.

Je remercie sincèrement M. Jean-Marie Pinon, Professeur des Universités à l'Institut Nationale des Sciences Appliquées de Lyon, pour avoir accepté d'être rapporteur de ce mémoire. Je le remercie également pour ses remarques pertinentes qui ont contribué à l'amélioration de la qualité de ce mémoire, ainsi que sa participation au jury.

Toute ma reconnaissance va à M. Gilles Zurfluh, Professeur à l'Université des Sciences Sociales, Toulouse I, avec lequel j'ai beaucoup appris, pour son soutien sans faille dans la direction de cette thèse et pour avoir guidé ce travail en conjuguant habilement disponibilité, conseils et critiques constructives tout en laissant libre court à ma créativité. Je lui suis très reconnaissante pour m'avoir permis de faire mes premiers pas dans le monde de la recherche et pour m'avoir aidé à poursuivre dans cette voie.

Je remercie sincèrement M. Claude Chrisment, Professeur à l'Université Paul Sabatier, Toulouse III, pour l'attention qu'il a portée à la lecture de ce mémoire. Je le remercie également pour ses remarques pertinentes qui ont contribué à l'amélioration de ce mémoire, ainsi que sa contribution au jury.

Je remercie sincèrement Mme. Chantal Soule-Dupuy, Professeur à l'Université des Sciences Sociales, Toulouse I, pour l'attention qu'elle a portée à la lecture de ce mémoire et pour ses remarques enrichissantes qui ont contribué à l'amélioration de ce travail. Je la remercie pour l'honneur qu'elle me fait en participant au jury.

Je présente toute ma gratitude à M. Franck Ravat, Maître de Conférences à l'Université des Sciences Sociales, Toulouse I, pour avoir suivi et encadré cette étude et pour sa collaboration ainsi que son aide précieuse. Je voudrais lui exprimer ma reconnaissance pour les efforts inépuisables qu'il a fournis, pour sa patience et sa disponibilité à tout moment malgré ses occupations. Je le remercie également pour l'honneur qu'il me fait en participant au jury.

Mes remerciements s'adressent également à M. Olivier Teste, Maître de Conférences à l'Université Paul Sabatier, Toulouse III, pour son encadrement continu tout au long de cette thèse, pour son aide précieuse,

ses judicieux conseils et sa contribution fructueuse dans la réalisation de ce mémoire et surtout pour sa bonne humeur et sa grande gentillesse. Je le remercie également pour avoir accepté de participer au jury.

Toute ma reconnaissance et ma gratitude pour M. Rafik Bouaziz, Maître Assistant à la Faculté des Sciences Economiques et de Gestion de Sfax, Tunisie, pour m'avoir mis sur la route de la recherche durant mon mémoire de maîtrise, pour avoir sacrifié une partie de ses vacances pour la lecture de ce mémoire et pour l'honneur qu'il me fait en participant au jury.

Je tiens à remercier mes enseignants de la Faculté des Sciences Economiques et de Gestion de Sfax pour m'avoir aidée à travailler avec l'équipe SIG d'une part et pour leurs conseils et aides d'autre part. Qu'ils trouvent ici l'expression de ma considération profonde.

Je tiens à remercier tous les membres de l'équipe SIG et plus particulièrement M. Mohand Boughanem pour sa confiance et son aide en ayant accepté d'être notre garant. Je remercie également Hamid Tebri « Inestimable ami », Mohamed Tmar, Anis Benammar, Kais Khrouf « Amis et futurs collègues » et Mohamed Mbarki « Tendre petit frère ».

Je souhaite remercier les personnes les plus proches, notamment une pensée bien particulière à mon adorable, cher et tendre époux, qui m'a aidé avec une indéfectible patience et a sacrifié son temps de travail pour que je puisse finaliser ma thèse. Un amour infini à ma petite poupée qui a dû supporter mon absence et le stress d'une thèse bien avant l'âge.

Je tiens à présenter toute ma reconnaissance à "Papa" et "Maman" qui m'ont permis de continuer mes études avec tout l'amour, la tendresse, la bonté et la fierté de cœurs parental et maternel.

Finalement, je tiens à remercier du fond du cœur, ma famille, mes sœurs Hanène et Zaineb, mes frères Houssem et Mohamed, mes beaux parents Ahmed et Monia, mes grands beaux parents Habib et Njaïba et mes beaux frères Imed, Nizar, Issam et le petit Yassin.

TABLE DES MATIERES

Chapitre III. Interrogation des données dimensionnelles sous contraintes

Chapitre IV. Méthode de conception d'un schéma dimensionnel contraint

Chapitre V. Outil d'aide à la conception de magasin dimensionnel contraint

INTRODUCTION

Contexte général de l'étude

L'accroissement du volume de données dans les systèmes d'information est de nos jours une réalité à laquelle chaque entreprise doit faire face. Notamment, elle doit permettre à ses responsables (« managers ») de déceler les informations pertinentes afin de prendre les bonnes décisions dans les plus brefs délais. Dans ce contexte, une enquête[1], réalisée par Teradata[2], estime que *"73 % des managers prennent de plus en plus de décisions dans le cadre de leur travail, 55 % disposent de moins en moins de temps pour prendre ces décisions et 54 % jugent que la quantité d'informations à traiter pour y parvenir augmente sans cesse"*. Pour faciliter la tâche de ces managers, les entreprises doivent définir des processus décisionnels reposant sur l'exploitation de nombreuses informations. Ces informations sont généralement présentes dans les systèmes d'information transactionnels. Néanmoins, l'exploitation de ces informations réparties et hétérogènes à des fins décisionnelles nécessite leur transformation sous une forme adaptée à l'analyse (Kimball et al, 2002).

Les systèmes décisionnels répondent à ces besoins en proposant des modèles et des techniques de manipulation des données. La plupart de ces systèmes repose sur **une approche OLAP** (« On Line Analytical Processing ») facilitant l'analyse interactive et la synthèse d'un grand volume de données (Codd et al, 1993).

Dans l'approche OLAP, les données sont souvent stockées dans des bases de données dimensionnelles, dans lesquelles les données sont organisées par centre d'intérêt et étudiées en fonction de différents axes d'analyse suivant un **modèle dimensionnel** (Kimball et al 2002).

Motivations et objectifs

Notre objectif est de proposer une solution complète permettant de concevoir, de modéliser et de manipuler des bases de données dimensionnelles assurant l'intégrité des données et la cohérence des restitutions.

Ces dernières années plusieurs propositions de modèles dimensionnels ont été faites (Gyssen et al, 1997) (Pedersen et al, 1999) (Trujillo et al, 2003). Ces travaux visent à proposer un modèle adapté aux besoins décisionnels s'adressant à des décideurs non informaticiens. Néanmoins, nous constatons le manque de modèles conceptuels permettant de faire abstraction des contraintes techniques de stockage de données et de mieux appréhender les besoins de ces décideurs (Golfarelli et al, 2002). Dans ce cadre, le modèle dimensionnel doit répondre à un ensemble de critères tels que le support de plusieurs sujets d'analyse facilitant la corrélation entre les données et l'expression explicite de perspectives multiples d'analyses au sein des axes d'analyse.

(Hûmmer et al 2002) dresse une liste de dix problématiques dans le domaine de modélisation dimensionnelle. Plusieurs de ces problématiques dénotent le besoin d'un mécanisme de contraintes dans le modèle dimensionnel. L'expression de ces contraintes permet, d'une part, de valider la structure dimensionnelle (structure hiérarchique des axes

[1] http://management.journaldunet.com/0401/040122_decision.shtml
[2] Teradata : Filiale de NCR (National Cash Register) Corporation spécialisée dans le stockage des données

d'analyse) et d'autre part de vérifier la sémantique des données analysées et notamment de désambiguïser les valeurs vides (ou nulles) qui peuvent provenir de la combinaison de perspectives d'analyses incompatibles (Carpani el al, 2001) (Hurtado et al, 2002).

En outre, le mécanisme de définition de contraintes au niveau du modèle dimensionnel doit être intégré dans le processus de manipulation des données dimensionnelles afin de reconnaître les combinaisons incohérentes et d'interdire leur visualisation. La définition d'un tel mécanisme permet de désambiguïser les analyses décisionnelles et d'assurer leur cohérence (Hümmer et al, 2002). Cependant, rares sont les travaux qui traitent de cette problématique au niveau des bases de données dimensionnelles (Hurtado et al, 2002).

Par ailleurs, l'importance du volume de données mis en jeu dans les systèmes d'aide à la décision nécessite des **mécanismes d'archivage** pour synthétiser l'information (Teste, 2000). Ces mécanismes offrent les moyens pour archiver les données temporelles à des niveaux de détail en adéquation avec les besoins des décideurs (Inmon, 1996). D'où l'importance de les intégrer au niveau du modèle dimensionnel.

Enfin, il est nécessaire d'offrir aux concepteurs aussi bien un modèle de données qu'une **démarche** et **un outil d'aide à la conception** de bases de données dimensionnelles contraintes. En effet, les méthodes classiques proposées dans le cadre des systèmes transactionnels ne permettent ni de prendre en compte les spécificités des applications décisionnelles ni du modèle dimensionnel. Or, cette problématique est peu étudiée dans la littérature (Hûmmer et al, 2002).

Contributions de nos travaux de recherche

Dans un contexte décisionnel, l'objet de nos travaux est de proposer un modèle conceptuel, un langage de manipulation et une méthode de conception des données dimensionnelles prenant en compte l'intégrité des données.

Le modèle dimensionnel que nous souhaitons proposer permet de représenter plusieurs centres d'intérêts étudiés en fonction de différentes vues ou axes d'analyse. Cette modélisation multi-centres d'intérêts et multi-vues permet de faciliter la mise en œuvre de corrélations lors des analyses décisionnelles. De plus, ce modèle supporte également l'expression de contraintes d'intégrité structurelles et sémantiques. Les contraintes structurelles servent à valider le schéma dimensionnel et notamment la hiérarchisation des axes d'analyse définie à l'aide de contraintes de dépendances fonctionnelles. Les contraintes sémantiques permettent d'intégrer les différentes règles de gestion de l'organisation.

Nos travaux doivent également compléter ce modèle par un langage algébrique d'interrogation de données dimensionnelles. L'intégration des contraintes dans ce langage assure la cohérence des données analysées en désambiguïsant les champs non renseignés (valeurs nulles) que le décideur ne peut interpréter. Elle permet, également, aux décideurs de préciser l'ensemble des instances à analyser.

Les contraintes interviennent, également, au niveau de la configuration de la base dimensionnelle basée sur la technique de matérialisation des vues. Cette technique consiste

à stoker les résultats des vues combinant les différents axes d'analyses représentant les requêtes possibles des décideurs afin de réduire le temps de réponse à ces requêtes. La nécessité d'un temps de rafraîchissement et d'un coût de stockage de ces vues engendre la problématique de sélection des vues à matérialiser. L'intégration des contraintes à ce niveau nous permet de réduire le nombre de vues candidates à la matérialisation.

Enfin, nous souhaitons proposer une méthode de conception adaptée pour la construction de bases de données dimensionnelles. Dans le cadre de cette méthode, nous souhaitons proposer une démarche de conception d'un schéma dimensionnel contraint intégrant l'ensemble des informations pertinentes à la prise de décision (besoins des décideurs et schéma des sources). De plus, cette méthode doit intégrer un outil d'aide à la conception intégrant une définition graphique des schémas conceptuels et une génération automatique de bases de données dimensionnelles.

Organisation du mémoire

Dans le premier chapitre, nous précisons le cadre d'étude de cette thèse en définissant les concepts de base des systèmes d'aide à la décision. Nous présentons un comparatif des travaux sur la modélisation dimensionnelle organisé par niveau d'abstraction : conceptuel, logique et physique. Nous étudions également l'intégration des contraintes sémantiques et structurelles dans le domaine de la modélisation dimensionnelle ainsi que la définition d'une méthode de conception de schéma dimensionnel. Enfin, nous présentons le contexte, la problématique et les objectifs de nos travaux dans ce domaine. Les chapitres deux à cinq présentent nos contributions.

Le deuxième chapitre présente les concepts inhérents à notre modèle dimensionnel en constellation (multi-centres d'intérêts) et les différentes contraintes structurelles et sémantiques intégrées dans ce modèle.

Le troisième chapitre étudie l'impact des contraintes sur l'interrogation de la base dimensionnelle à deux niveaux. Au niveau de la manipulation, il propose un langage algébrique de manipulation des données dimensionnelles supportant les contraintes. Ce langage propose d'étendre les opérateurs dimensionnels afin de préciser l'ensemble des données dont le décideur a besoin. Au niveau de l'optimisation de la manipulation, le chapitre décrit notre processus de sélection des vues intégrant les contraintes.

Le quatrième chapitre décrit notre méthode de conception des bases de données dimensionnelles contraintes basées sur la dichotomie d'espace de stockage entrepôt et magasin de données. Nous proposons une méthode mixte intégrant l'expression des besoins décideurs suivant une démarche descendante et la description des données sources en appliquant une démarche ascendante. En outre, notre méthode intègre l'expression de contraintes au niveau de la modélisation du schéma dimensionnel.

Le cinquième chapitre est consacré à la présentation de notre outil d'aide à la conception graphique de bases de données dimensionnelles (GMAG : Générateur de MAGasins de données dimensionnelles). Cet outil permet d'assister le concepteur lors de la construction graphique des magasins de données dimensionnelles à partir d'un entrepôt de données historisées.

CHAPITRE I : CONTEXTE DE L'ETUDE

PLAN DU CHAPITRE

Face à la mondialisation des échanges et à la concurrence accrue, les dirigeants des entreprises ont besoin d'avoir une vision claire de leur environnement. Cette vision est fournie par des outils faciles à utiliser et qui ne perturbent pas le système opérationnel. Chaque entreprise souhaite exploiter intelligemment ses données et avoir plus d'informations que ses concurrents afin de pouvoir prendre les décisions les plus efficaces. D'où l'ère de l'informatique décisionnelle.

Dans ce chapitre, nous introduisons dans une première section l'approche de l'aide à la décision, notamment, les concepts dimensionnels dans le contexte des systèmes décisionnels. Une deuxième section présente les travaux relatifs à la modélisation dimensionnelle suivant les différents niveaux d'abstraction (conceptuel, logique et physique) et un comparatif de ces travaux. Une troisième section décrit la typologie des contraintes exprimées dans les modèles dimensionnels afin de conserver l'intégrité des données et de conserver la cohérence de l'analyse. Nous exposons, ensuite, l'état de l'art des méthodes de conception des schémas dimensionnels. Enfin, nous présentons notre proposition dans le contexte des systèmes décisionnels.

1. L'aide à la décision

L'aide à la décision a pour objectif d'accompagner un ou plusieurs décideurs dans le processus de prise de décision. Elle permet aux acteurs concernés de spécifier leurs besoins par des processus de collecte, d'analyse et d'échange d'informations (Kimball et al, 2002).

Dans cette section, nous décrivons les caractéristiques des systèmes décisionnels d'une manière générale en spécifiant les caractéristiques des systèmes OLAP. Dans ce contexte, notre équipe a proposé une architecture basée sur l'approche des entrepôts et des magasins de données décrite dans le troisième paragraphe. Enfin, nous présentons le modèle dimensionnel dédié à l'aide à la prise de décision et les opérateurs de manipulation des données de ce modèle.

1.1. Caractéristiques des systèmes décisionnels

Les systèmes d'information se sont souvent développés par domaine d'activité : financier, commercial, marketing, etc. L'information accumulée est très diverse et elle est gérée par des systèmes hétérogènes (Gardarin, 1999). Le but de ces systèmes est de fournir aux organismes l'infrastructure nécessaire pour réaliser leurs tâches quotidiennes.

Un grand besoin d'intégration de ces systèmes, dit transactionnels « OLTP : On-Line Transactional Processing », est ressenti afin de permettre à tous les acteurs de disposer des informations relatives à leurs centres d'intérêts (Gardarin, 1999). Ces informations doivent pouvoir être accessibles et faciles à interroger par le décideur en fonction de son secteur d'activité (marketing, économique, ...) (Codd et al, 1993) (Kimball et al, 2002).

L'approche adoptée pour répondre à ce besoin est de regrouper les informations disparates, après les avoir pré-traitées, au sein d'un unique espace de stockage de données intégrées par sujet. L'analyse de ces données par des requêtes interactives devient alors possible et permet de prendre rapidement de meilleures décisions. Différents outils d'analyse peuvent être greffés sur cet espace tels que les outils d'analyse interactive, les outils de fouille de données permettant l'extraction de nouvelles connaissances et les requêteurs fournissant des tableaux de bord aux différents acteurs de la décision.

Ainsi, le système décisionnel obtenu est basé sur deux composantes, un espace de stockage de données de synthèse, intégrées et historisées et des outils d'analyse qui assurent la présentation des données à l'aide d'interfaces graphiques.

Définition

Un système décisionnel est un système d'information qui regroupe les données d'aide à la décision et facilite leur exploitation en fournissant les outils adéquats.

Dans le contexte des systèmes décisionnels, nous avons étudié les systèmes OLAP « On-Line Analytical Processing » (Codd et al, 1993) qui proposent de :

- collecter les données pertinentes,
- les organiser selon des structures adaptées à la prise de décision,
- les interroger d'une manière interactive et dynamique.

Une plus ample description de ces systèmes est présentée dans la section suivante.

1.2. Systèmes OLAP

Dans la littérature, plusieurs définitions sont proposées pour les systèmes OLAP (Codd et al, 1993) (Villacampa, 2002) (OLAP Report1). Dans ces définitions, les caractéristiques de base sont la structure dimensionnelle des données, les données forment des points dans un espace à plusieurs dimensions, et l'interactivité de l'interrogation afin de s'approcher de la perception du décideur et de l'aider au mieux dans son processus de prise de décision. Nous proposons alors de définir les systèmes OLAP comme suit :

Définition

Un système OLAP est un système d'information décisionnel qui organise les données dans un espace dimensionnel. Il regroupe un ensemble d'outils en interaction qui réalisent la synthèse dynamique, l'analyse interactive et l'agrégation d'un grand volume de données afin d'améliorer le processus de prise de décision.

Ce système réunit un ensemble de nouvelles fonctionnalités décrites par 12 règles (Codd et al, 1993). Les principales caractéristiques extraites de ces règles sont :

- la vision dimensionnelle des données, la transparence entre l'outil de visualisation et l'espace de stockage des données dimensionnelles,
- l'interopérabilité (l'outil rend invisible à l'utilisateur l'hétérogénéité des données),
- la manipulation intuitive des données et la flexibilité des restitutions, le décideur dispose d'une interface ergonomique de consultation.

Les systèmes OLAP visent à combler les lacunes des systèmes transactionnels. En effet, une des principales caractéristiques des systèmes transactionnels, est une activité de modification et d'interrogation fréquentes et répétitives (Kimball et al, 2002). L'accès au système est réalisé par de très courtes transactions. Enfin, la plupart de ces systèmes ne conservent pas les évolutions des données manipulées, seules les versions courantes sont conservées.

[1] OLAP Report: http://www.olapreport.com/. C'est une ressource indépendante de recherche pour des organismes achetant et mettant en ligne des applications OLAP.

Contrairement aux systèmes OLTP, les utilisateurs des systèmes OLAP n'ont aucun besoin de modification des données analysées. Ces utilisateurs ont besoin d'outils interactifs et simples, supportant la prise de décision. Ces besoins s'articulent souvent autour d'un métier particulier (marketing, finance, …) et nécessitent de répondre à des requêtes ad hoc (non prévues par le système) et complexes (agrégation, intégration). L'aide à la décision nécessite aussi de conserver l'historique des données afin d'anticiper les décisions futures.

Le tableau suivant compare les caractéristiques des systèmes OLAP et OLTP.

	OLTP	**OLAP**
Utilisateur	**Agents opérationnels, nombreux**	**Analystes, décideurs, peu nombreux**
Fonctionnalité	**Opérations journalières**	**Support de décision**
Modèle de données	**Orienté application**	**Orienté sujet**
Données	**Actuelles, mises à jour, détaillées, plates et isolées**	**Historisées, agrégées, intégrées, dimensionnelles, consolidées**
Usage	**Répétitif**	**Ad hoc**
Accès	**Lecture/ écriture**	**Lecture**
Requêtes	**Simples**	**Complexes**

***Tableau I.1** : COMPARAISON DES PROCESSUS OLTP ET OLAP*

Dans la littérature, nous retrouvons souvent le mot OLAP associé à l'approche des entrepôts de données. Cette approche représente un axe de recherche dans le contexte des systèmes décisionnels. La section suivante présente les caractéristiques de cette approche.

1.3. Entrepôts et magasins de données

Les entrepôts de données constituent une solution adéquate pour construire un système *décisionnel* (Widom, 1995) (Inmon, 1996). Un entrepôt de données est défini comme étant « *une collection de données intégrées, orientées sujet, non volatiles, historisées, résumées et disponibles pour l'interrogation et l'analyse* » (Inmon, 1996). Cette définition met l'accent sur les caractéristiques suivantes :

Intégrées : Les données alimentant l'entrepôt proviennent de sources multiples et hétérogènes. Les données des systèmes de production doivent être converties, reformatées et nettoyées de façon à avoir une seule vision globale dans l'entrepôt.

Orientées sujet : Les données s'organisent par sujets ou thèmes, contrairement aux données des systèmes de production généralement organisées par processus fonctionnel. L'intérêt de cette organisation est de disposer de l'ensemble des informations utiles sur un sujet, le plus souvent transversal aux structures fonctionnelles et organisationnelles de l'entreprise.

Non volatiles et ***historisées :*** Les données des systèmes opérationnels sont constamment manipulées, modifiées ; elles sont mises à jour à chaque nouvelle transaction. Par opposition, les données de l'entrepôt sont le reflet d'un instantané des données du système opérationnel. Lorsqu'intervient un changement important dans les données, une nouvelle photo est prise de façon à ce que l'entrepôt garde une trace de l'historique des données.

Une architecture générale du système décisionnel, basée sur l'approche des entrepôts de données, est présentée dans la Figure I.1.

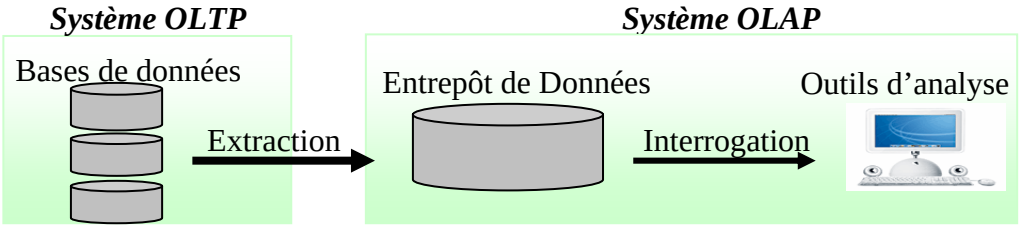


Figure I.1 : *APPROCHE DES ENTREPOTS DE DONNEES*

Dans l'architecture précédente, un seul espace de stockage est défini pour les données décisionnelles : l'entrepôt de données doit permettre de recueillir, stocker et intégrer un grand volume de données centralisées et, en même temps, de répondre à des requêtes des utilisateurs concernant un thème, un métier ou une analyse spécifique. Nous distinguons là deux problématiques indépendantes : (1) la gestion efficace des données "historisées", "centralisées" (intégration des sources) et (2) la définition d'un sous ensemble de données autour d'un thème particulier afin de répondre aux besoins spécifiques de ses utilisateurs. Aussi, l'architecture des systèmes décisionnels que nous élaborons est basée sur une dichotomie d'espaces de stockage : l'entrepôt et les magasins de données (Ravat et al, 2000a).

Définition

*L'**entrepôt** est le lieu de stockage centralisé et extrait des sources. Il intègre et «historise » l'ensemble des données utiles pour les prises de décisions. Son organisation doit faciliter la gestion des données et la conservation des évolutions.*

*Chaque **magasin** est un extrait de l'entrepôt. Les données extraites sont adaptées à un groupe de décideurs ou à un usage particulier. L'organisation des données doit suivre un modèle spécifique qui facilite les traitements décisionnels.*

Dans la Figure I.2, nous schématisons l'architecture des systèmes décisionnels tel que nous l'avons définie précédemment (Ravat et al, 2000b).

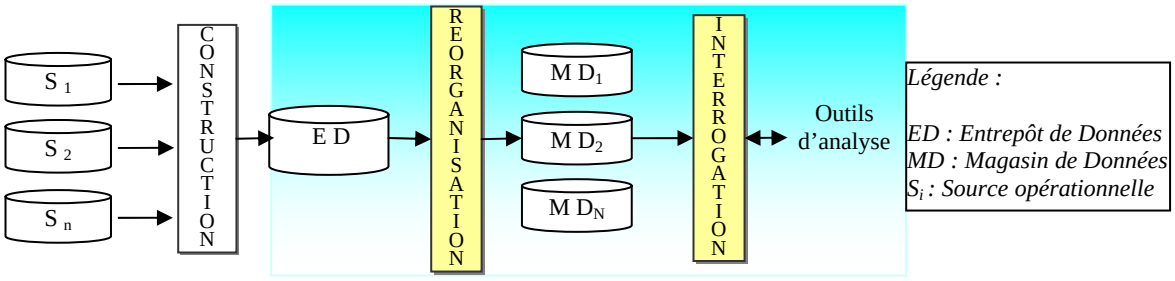


Figure I.2 : *ENTREPOT ET MAGASINS DE DONNEES.*

La construction consiste à extraire les données pertinentes pour la prise de décision et à les recopier dans l'entrepôt de données. Celui-ci constitue une collection centralisée de données matérialisées et historiques (Baril et al, 2003). Le modèle de l'entrepôt doit supporter des structures complexes (Pedersen et al, 1998) et supporter l'évolution des données (Pedersen et al, 1999) (Yang et al, 2000) (Teste, 2000) (Bellahsène, 2002) (Mendelzon et al, 2003).

La réorganisation permet de restructurer les données entreposées en les stockant dans des magasins de données visant à supporter efficacement les processus d'interrogation et d'analyse (Ravat et al, 2001).

L'interrogation consiste à utiliser les données des magasins pour prendre des décisions. La représentation des données doit faciliter leur compréhension et leur manipulation par les décideurs non informaticiens (tableaux à n dimensions, graphiques, …).

Nos travaux se focalisent sur les deux dernières étapes permettant la réorganisation et l'interrogation des magasins de données décisionnelles. Notre approche est basée sur un modèle dimensionnel de données. Nous présentons les concepts de base de ce modèle dans la section suivante.

1.4. Concepts de la modélisation dimensionnelle

Le modèle dimensionnel répond aux lacunes des modèles transactionnels. Il vise à présenter les données sous une forme intuitive dont l'objectif est de se rapprocher de la manière dont les décideurs perçoivent les données d'analyse (Codd et al, 1993) (Kimball et al, 2002). Ce modèle propose de visualiser les données représentant les sujets d'analyse comme des points dans un espace à plusieurs dimensions formant les différents axes d'analyse (Choong et al, 2003).

Définition

*La **modélisation dimensionnelle** considère les données comme des points dans un espace à plusieurs dimensions. Ces points représentent les centres d'intérêts décisionnels (sujets) analysés en fonction des différents axes d'analyse.*

Le modèle dimensionnel est basé sur la dualité des concepts fait - dimension (Kimball et al, 2002).

Un ***fait*** représente un sujet d'analyse dans une application décisionnelle. Supposons, par exemple, que nous souhaitons analyser les performances des agences dans une société de location de véhicules. Dans un schéma dimensionnel, ce besoin est modélisé par le ***fait*** *Location.*

Définition

*Un **fait** est un centre d'intérêt décisionnel. Il regroupe un ensemble d'attributs numériques représentant les mesures d'activité.*

Afin de calculer la performance des agences, nous définissons les ***mesures*** d'activités montant et durée des locations dans le fait *Location*.

Définition

*Une **mesure** est un indicateur d'analyse de type numérique et cumulable. Une mesure est accompagnée d'un ensemble de fonctions d'agrégation qui permettent de l'agréger en fonction des axes d'analyse.*

Les mesures sont réunies dans un même fait si elles peuvent être analysées suivant les mêmes axes d'analyse. Les faits comportent un très grand volume de données pouvant être résumées, lors des interrogations, grâce aux opérations d'agrégation (somme, moyenne, max, min, ...) (Kimball et al, 2002). Or, ces opérations ne peuvent être appliquées que sur des données numériques et additives.

Nous souhaitons analyser les mesures du fait *Location* en fonction des agences. La définition d'une ***dimension*** qui regroupe les données relatives à une agence, permet de répondre à ce besoin.

Définition

*Une **dimension** est un axe d'analyse selon lequel sont visualisées les mesures d'activité d'un sujet d'analyse.*

Parmi les attributs d'une dimension, nous retrouvons les paramètres de l'analyse. Par exemple, l'analyse du fait *Location* est réalisée en fonction de la dimension *Agence* aux niveaux du code, de la ville, de la région ou du pays de l'agence. Ces attributs représentent les paramètres d'analyse de la dimension *Agence*.

Définition

*Un **paramètre** est un attribut appartenant à une dimension. Il représente un niveau de détail selon lequel sont visualisées les mesures d'activité d'un sujet d'analyse.*

Les paramètres peuvent être accompagnés de descripteurs appelés ***attributs faibles*** (Teste, 2000). Par exemple, l'identifiant d'une agence *Code_Ag* peut être accompagné par le nom de celle-ci. L'ensemble composé du paramètre et de ses attributs faibles est appelé ***niveau hiérarchique***.

Définition

*Un **attribut faible** est un descripteur de paramètre. Cet attribut n'est pas utilisé dans les calculs de regroupement lors des opérations d'agrégation ; il a un rôle informationnel permettant de faciliter les analyses.*

Les paramètres d'une dimension sont organisés en une ou plusieurs ***hiérarchies***, de la granularité la plus fine vers la granularité la plus générale. Par exemple, les paramètres de la dimension *Agence* sont organisés suivant la hiérarchie géographique de la granularité *Code_Ag* vers la granularité *Ville, Région,* puis *Pays*. Les hiérarchies sont primordiales dans le modèle dimensionnel puisqu'elles sont employées pour manipuler les mesures lors des opérations d'agrégation. Le changement de paramètre d'une hiérarchie implique le changement de la granularité ; par exemple, le regroupement des montants des locations en fonction du paramètre *Ville* puis en fonction du paramètre *Pays* selon la dimension *Agence* permet de passer d'une analyse par ville vers une analyse par pays des locations.

Définition

*Une **hiérarchie** est une perspective d'analyse définie dans une dimension. Elle regroupe un ensemble de paramètres organisés de la granularité la plus fine vers la granularité la plus générale.*

La combinaison de ces différents concepts permet de construire des schémas en étoile ou en constellation (Teste, 2000) (Moody et al, 2000).

1.4.1. Schéma en étoile

Dans un tel schéma, les mesures sont regroupées dans un seul fait relié à plusieurs dimensions regroupant les paramètres de l'analyse.

Exemple 1

Cet exemple vise à définir un magasin dimensionnel permettant d'analyser le montant et la durée des locations de véhicules selon trois axes *Agence, Véhicule et Temps*. Une agence est caractérisée par son code, son nom et sa localisation décrite par les informations ville, région et pays. Un véhicule est caractérisé par son immatriculation, sa marque, sa catégorie et son type de moteur. Au niveau de l'axe temps, nous souhaitons avoir les montants et les durées des locations journalières, mensuelles et annuelles. Pour répondre à ces besoins, nous avons défini le fait *Location* de véhicule, comportant les mesures *Mt_Loc* et *Durée_Loc*, analysé selon les dimensions *Agence*, *Temps* et *Véhicule* (cf. Figure I.3).

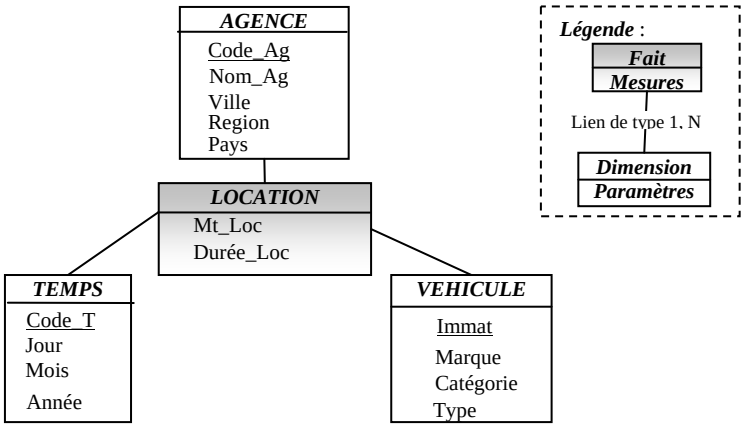


***Figure I.3** : EXEMPLE D'UN SCHEMA EN ETOILE (KIMBALL ET AL, 2002)*

Comme le montre la Figure I.3, cette représentation du schéma en étoile n'explicite pas les hiérarchies des paramètres.

Remarque : Cet exemple servira de base à la présentation des différents travaux relatifs à la modélisation dimensionnelle présentés à la section 2.

1.4.2. Schéma en constellation

Ce schéma est une extension du schéma en étoile (cf. Figure I.4). Il consiste à fusionner plusieurs schémas en étoile qui utilisent des dimensions communes. Un schéma en constellation comprend donc plusieurs faits reliés à un ensemble de dimensions qui peuvent être partagées.

Ce schéma présente l'avantage de pouvoir corréler les sujets d'analyse tels que la comparaison des montants des locations réalisées dans les différentes agences par rapport aux chiffres d'affaires réalisés par son personnel. En outre, le partage des dimensions par plusieurs faits permet d'éviter de les définir plusieurs fois.

Exemple 2

Nous souhaitons comparer les performances des agences en terme de montant et de durée de location avec les performances de ses employées (chiffres d'affaires et marge). Ce besoin est présenté par l'ajout du fait *Performance* à notre schéma en étoile de l'exemple 1. Ce nouveau fait est analysé en fonction des dimensions *Employé, Agence* et *Temps.*

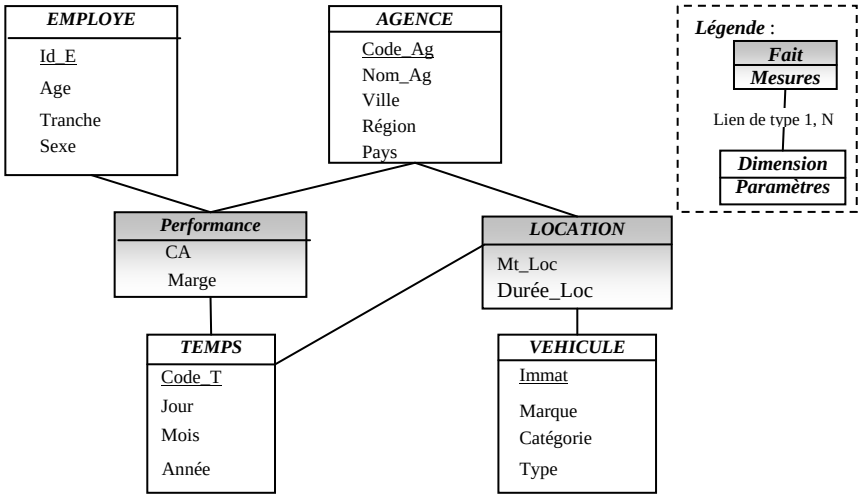


***Figure I.4** : EXEMPLE D'UN SCHEMA EN CONSTELLATION*

1.4.3. Bilan

Le modèle dimensionnel organise les données d'une manière adaptée aux analyses et vise à aider les décideurs non informaticiens lors de la prise de décision. Ce modèle est représenté par un schéma en étoile ou en constellation. Le premier est composé d'un seul fait (sujet d'analyse) analysé en fonction des différentes dimensions (axes d'analyse dont les paramètres sont organisés en hiérarchies multiples). Le deuxième regroupe plusieurs faits reliés à plusieurs dimensions qui peuvent être partagées.

Cette structure dimensionnelle est souvent accompagnée par une représentation sous forme de cube de données (Codd et al, 1993) visant à faciliter la manipulation des données décisionnelles. Pour répondre à ce besoin, un ensemble d'opérateurs dimensionnels est proposé aux décideurs. Nous décrivons dans la section suivante les différentes opérations de manipulation dimensionnelle.

1.5. Manipulation dimensionnelle

Les données dimensionnelles sont représentées au travers d'un ***cube*** regroupant à la fois la structure et les valeurs des données (voir Figure I.5). Chaque case dans le cube présente les valeurs des mesures d'un fait (par exemple les montants des locations sont représentées à l'intersection des dimensions *Agence*, *Véhicule* et *Temps*). Chaque arête du cube, représentant une dimension, est composée des valeurs d'un paramètre de la dimension considérée.

La Figure I.5 présente le cube analysant les mesures du fait *Location* en fonction des paramètres *Année* de la dimension *Temps*, *Marque* de la dimension *Véhicule* et *Ville* de la dimension *Agence*.

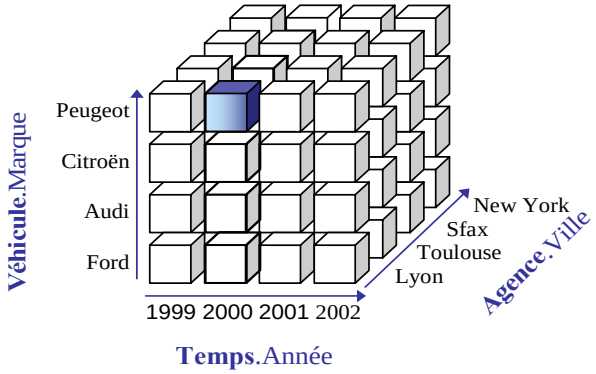


***Figure I.5** : EXEMPLE DE CUBE DIMENSIONNEL*

Une nouvelle génération d'opérateurs algébriques basés sur le concept de cube a vu le jour (Codd et al, 1993). La représentation dimensionnelle fait appel à des opérateurs spécifiques qui faciliteront l'analyse et la visualisation des cubes dimensionnels.

L'opérateur de rotation (**«Rotate»**) permet d'avoir accès aux différentes vues de données : c'est le fait d'inverser les axes visualisés du cube. Un cube de n dimensions possède n * (n – 1) vues possibles.

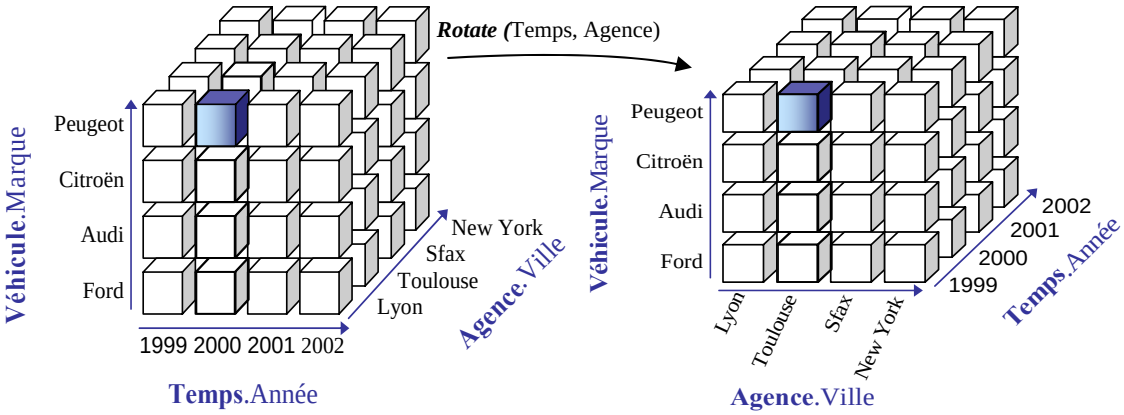


***Figure I.6** : ROTATION DES DIMENSIONS AGENCE ET VEHICULE*

Cette rotation du cube, nous permet de visualiser les locations en fonction des marques de véhicules et des villes. Les locations en fonction des années passent au plan latéral.

Les opérateurs de restriction (***«Slice and Dice»***) permettent de restreindre les valeurs dans le cube. Slice est appliqué sur les dimensions tel que restreindre l'analyse des ventes aux villes de Toulouse et de Lyon (Figure I.7). Dice sert à restreindre les données des faits.

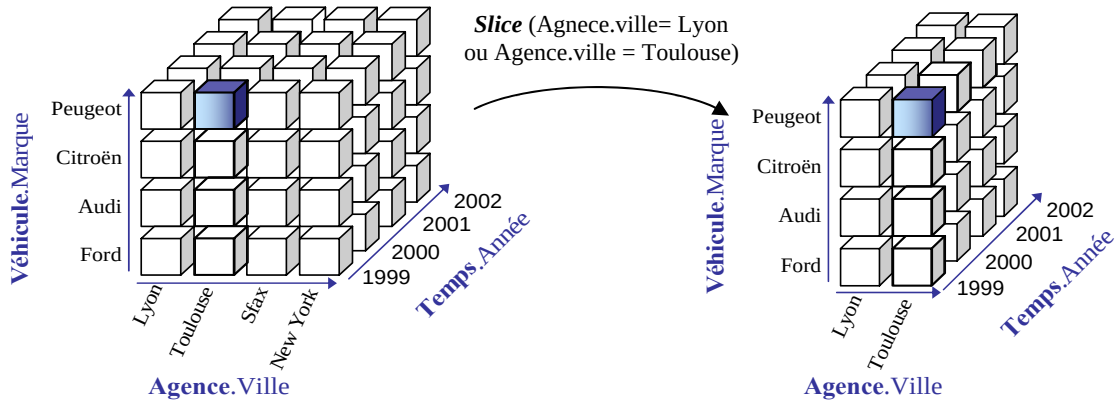


Figure I.7 : *L'OPERATEUR DE RESTRICTION SLICE*

Les opérateurs de forage vers le haut et de forage vers le bas (**«Roll_Up»** et **«Drill_Down»**) permettent soit de généraliser l'analyse, soit de l'affiner en modifiant le paramètre utilisé pour définir les valeurs d'une arête du cube. En effet, les dimensions sont associées à des hiérarchies ; ces deux opérateurs permettent, respectivement, de « monter » ou de « descendre » dans une hiérarchie.

Dans l'exemple de la Figure I.8, nous présentons une opération de forage vers le bas à partir du paramètre *Ville* vers le paramètre *Code_Ag*.

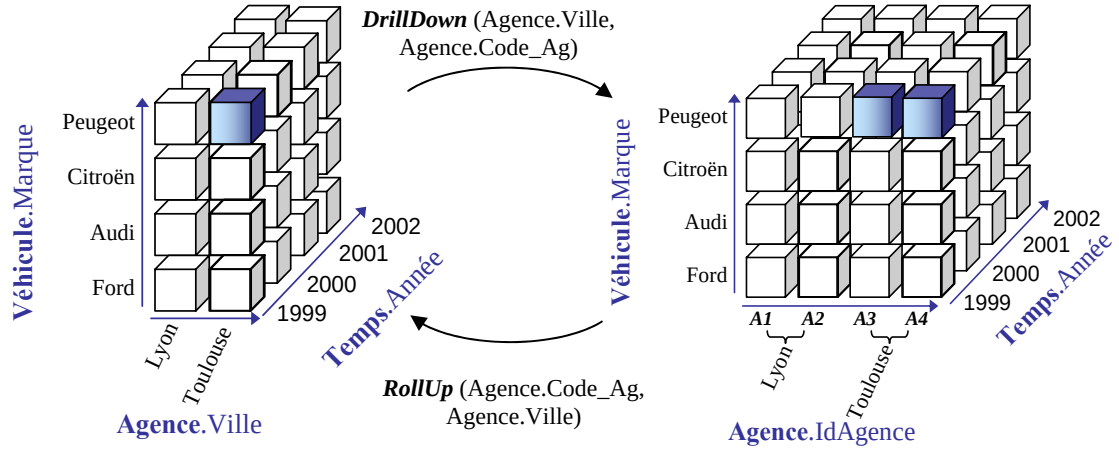


Figure I.8 : *LES OPERATEURS DE FORAGE*

Nous constatons que malgré le nombre important de travaux sur la manipulation, aucun standard de langage de manipulation dimensionnelle n'est unanimement accepté. Nous consacrons la section suivante à la présentation de ces travaux.

2. Modélisation des données dimensionnelles : Etat de l'art

Plusieurs modèles dimensionnels ont été proposés dans la littérature. Nous présentons dans ce qui suit une classification de ces modèles en se basant sur les trois niveaux d'abstraction (conceptuel, logique et physique) :

- Le niveau conceptuel répond à la question '*quoi*' et se restreint à décrire la vision de l'utilisateur des données indépendamment de l'implantation.
- Le niveau logique décrit le '*comment*'. Autrement dit, ce niveau transforme les concepts du niveau conceptuel en tenant compte d'un modèle de données particulier.

– Le niveau physique décrit les techniques utilisées pour l'implantation du modèle logique en tenant compte des spécificités d'une plate forme particulière.

Pour le niveau conceptuel, nous proposons d'organiser les modèles dimensionnels en trois catégories selon le paradigme utilisé : les modèles qui étendent le modèle Entité Association, les modèles qui se basent sur le paradigme objet et les modèles qui présentent un modèle conceptuel purement dimensionnel. Parallèlement, au niveau logique, trois axes ont dominé les recherches : le ROLAP basé sur le modèle relationnel, le OOLAP manipulant des objets et le MOLAP qui gère des vecteurs dimensionnels.

Au niveau physique, plusieurs travaux se sont focalisés sur les techniques supportant les fonctionnalités OLAP. Notamment, les techniques de matérialisation des vues, d'indexation optimisée (binaire, jointure en étoile, …), de réécriture de requêtes, d'exécution en parallèle de ces requêtes, etc. Nous présentons en particulier les deux techniques les plus étudiées dans les systèmes OLAP : les approches des vues matérialisées et d'indexation optimisée.

2.1. Niveau conceptuel

Les propositions de modèles conceptuels pour les données dimensionnelles sont relativement récentes dans ce domaine. Dans les sections suivantes, nous présentons les travaux qui se basent sur des modèles conceptuels classiques représentés par le paradigme Entité-Association et le paradigme objet et ceux qui se basent sur un modèle purement dimensionnel.

2.1.1. Extension des modèles existants

Plusieurs auteurs ont choisi de baser leurs modèles dimensionnels sur une approche existante, notamment le modèle Entité - Association et le modèle orienté objet. Ce choix vise à profiter de la popularité de ces modèles et de la maturité et la robustesse de leurs concepts et formalismes.

2.1.1.1. Paradigme Entité - Association

Les modèles dimensionnels qui se basent sur le paradigme Entité-Association, gardent les mêmes formalismes de ce paradigme pour exprimer les nouveaux concepts dimensionnels. L'extension minimale du modèle Entité-Association permet de tirer profit de sa popularité et de faciliter l'acceptation du nouveau modèle dimensionnel par les concepteurs.

♦ ***StarER***

(Tryfona et al, 1999) propose un modèle conceptuel pour les données dimensionnelles appelé StarER. Ce modèle combine les concepts du modèle Entité - Association et les concepts dominants de la modélisation dimensionnelle.

Ce modèle se base sur quatre concepts. Le ***fait*** est un nouveau concept intégré dans le modèle Entité-Association. Il est représenté, graphiquement, par un cercle (ex. Location véhicule dans la Figure I.9). Le concept d'***entité*** modélise les niveaux hiérarchiques des dimensions de l'analyse et regroupe un ensemble de propriétés représentant les attributs faibles. Une ***entité*** est représentée par un rectangle (ex. Agence, Ville, Région, Pays, …). Le concept de ***relation*** permet la modélisation des liens entre les entités ou entre les entités

et les faits. La cardinalité d'une relation peut être de type M:N, N:1 ou 1:N. Par exemple, la relation entre les entités *Ville* et *Région* est de cardinalité 1 : N indiquant qu'une ville appartient à une seule région et qu'une région contient plusieurs villes. Les relations entre les entités peuvent être de type spécialisation/généralisation, agrégation ou composition. Le concept d'***attribut*** représente les propriétés des entités ou des faits. Au niveau graphique, un attribut est représenté par un ovale.

Exemple 3

L'exemple d'analyse dimensionnelle que nous avons présenté est composé du fait *Location* et des dimensions *Véhicule*, *Temps* et *Agence*.

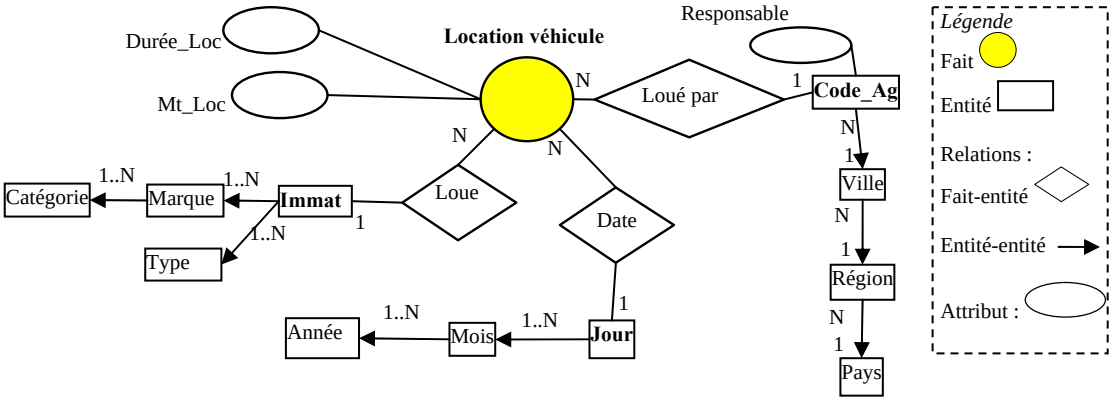


Figure I.9 *: MODÈLE STARER (TRYFONA ET AL, 1999)*

La complexité de ce modèle et le souci de représenter tous les détails inutiles pour une application OLAP rend difficile son exploitation et sa compréhension par des utilisateurs du monde de l'entreprise. Ce modèle ne propose pas d'opérateurs spécifiques à la manipulation des données dimensionnelles. En effet, il utilise les opérateurs relationnels classiques.

♦ ***Modèle Entité-Association Dimensionnel***

(Sapia et al, 1999) (Hahn et al, 2000) proposent une extension du modèle Entité-Association afin de pouvoir définir les concepts dimensionnels en respectant les deux principes suivants :

– une extension minimale de ce modèle,

– la représentation de la sémantique dimensionnelle.

En suivant ces principes, (Sapia et al, 1999) propose un ensemble d'extensions sous forme de spécialisations (relation d'héritage) des concepts Entité-Association. Ainsi, un niveau hiérarchique est représenté par une spécialisation du concept ***Entité*** schématisée par un rectangle. Par exemple, le niveau hiérarchique décrivant le code d'une agence est décrit par l'entité *Code_Ag*. Cette entité comporte un attribut faible représentant le responsable de l'agence. Les dimensions de l'analyse sont composées d'un ensemble de niveaux hiérarchiques reliés par des associations binaires appelées « ***Rollup-to*** », comme par exemple, la relation entre les niveaux *Mois* et *Année*. Cette relation définit un graphe acyclique direct entre les niveaux hiérarchiques. Le ***fait*** est modélisé par une association n-aire reliant les niveaux hiérarchiques les plus détaillés représentant les différentes dimensions.

Exemple 4

La Figure I.10 présente un exemple basé sur la représentation graphique du modèle dimensionnel ME/R proposé par (Sapia et al, 1999).

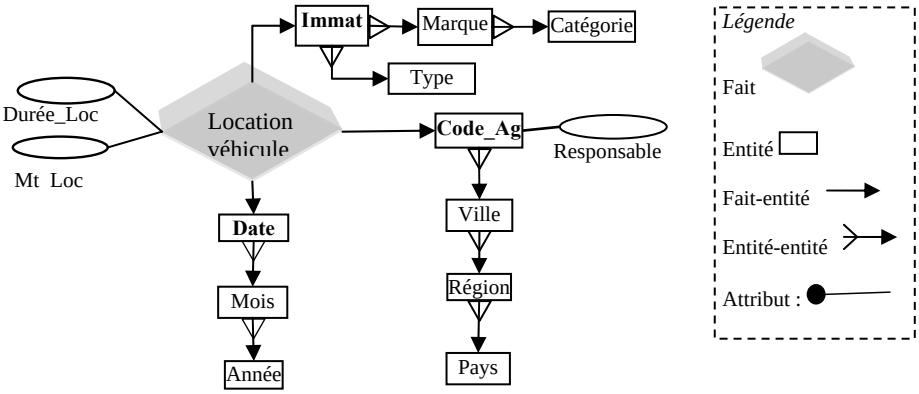


***Figure I.10** : MODELE ME/R (SAPIA ET AL, 1999)*

Le modèle propose un ensemble d'extensions du modèle Entité-Association sous forme de spécialisations. Ces extensions comportent une définition explicite des niveaux hiérarchiques ; les hiérarchies sont donc représentées d'une manière explicite.

♦ ***Bilan des modèles Entité-Association***

Les modèles dimensionnels Entité-Association proposent d'étendre les concepts d'entité et d'association pour exprimer les faits et les dimensions. Une dimension est généralement représentée par un ensemble d'entités. Chaque entité correspond à un niveau hiérarchique et les différents niveaux sont reliés par des associations binaires. Les faits correspondent à des associations entre les différentes dimensions. Ces modèles tirent profit de la popularité du modèle Entité-Association et de sa force d'expressivité pour faciliter la compréhension et la conception du modèle dimensionnel. Par contre, ces modèles restent complexes et nécessitent la maîtrise d'un ensemble de concepts informatiques tels que les concepts d'entité, de relation et d'identifiant, qui ne font pas partie du vocabulaire décisionnel et restent difficiles à appréhender par un décideur non informaticien. En outre, ces modèles n'expriment pas les contraintes qui existent entre les données dimensionnelles.

2.1.1.2. Paradigme objet

Les modèles dimensionnels objet, se basent sur les concepts de classe et d'objet. Dans ces modèles, les concepts d'héritage, de composition et d'agrégation sont souvent utilisés pour définir les relations entre les différents concepts dimensionnels.

♦ ***GOLD***

(Trujillo et al, 2003) et (Lujan et al, 2004) décrivent un modèle conceptuel orienté objet basé sur une extension du diagramme de classe UML.

Le fait présenté par une classe d'association est décrit par un ensemble d'attributs appelés mesures. Le fait est relié aux dimensions (représentées elles aussi par des classes UML) au travers d'une relation d'agrégation. Ces dimensions sont composées d'un ensemble d'attributs. Les hiérarchies sont définies comme un graphe acyclique de niveaux hiérarchiques modélisés par des classes. Le modèle supporte la multi-hiérarchisation au

sein des dimensions. Il exprime la contrainte de «*complétude*» sur les instances des hiérarchies ; une hiérarchie satisfait cette contrainte si elle comporte toutes les instances de la dimension.

Exemple 5

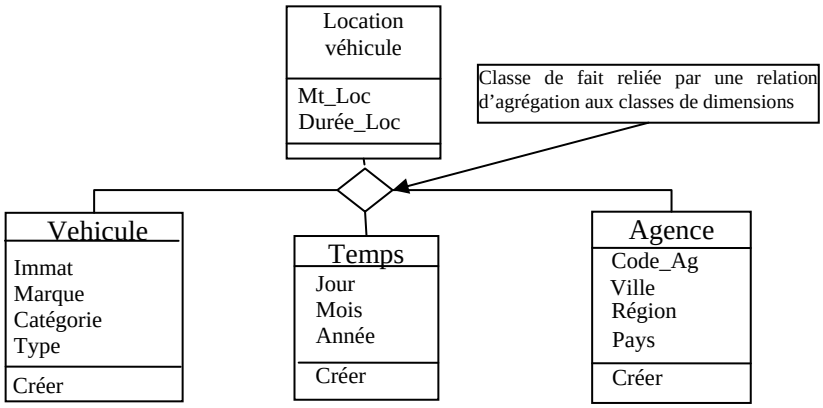


***Figure I.11** : SCHEMA MULTIDIMENSIONNEL (TRUJILLO ET AL, 2003)*

La caractéristique de complétude définie sur les classes des niveaux hiérarchiques permet d'exprimer partiellement les contraintes sémantiques. Elle ne permet pas par exemple, de traduire une exclusion entre les instances de deux hiérarchies.

♦ ***EMDM: Extended Multidimensional Data Model***

(Pedersen et al, 1999) (Pedersen et al, 2004b) proposent un modèle dimensionnel basé sur le paradigme orienté objet. Ce modèle fournit un formalisme et une algèbre qui répondent aux besoins d'aide à la décision recensés dans le contexte médical.

Le schéma dimensionnel de fait proposé est défini par le couple ***S = (F, D)*** avec ***F*** une classe de fait et **D** l'ensemble des dimensions reliées à ce fait. Une *dimension* est représentée au travers d'un ensemble ordonné de niveaux hiérarchiques, appelés *catégories*, organisés du niveau le plus détaillé noté $\perp$, au niveau le plus général noté $\top$. Le modèle définit l'ensemble des fonctions d'agrégation applicables à un niveau donné. Les instances du *fait* sont des objets de la classe **F**. La combinaison des valeurs des paramètres de plus bas niveau des dimensions caractérise un objet du fait mais ne forme pas son identifiant. Par contre, l'objet du fait lui-même n'est pas dupliqué dans la classe.

Le modèle traite les faits et les dimensions d'une manière symétrique ; les faits font partie de l'ensemble des dimensions (cf. Figure I.12). Il permet, également, de définir des hiérarchies multiples par dimension d'analyse.

Exemple 6

Dans notre exemple, nous remarquons que la mesure montant de location (*Montant)* est gérée comme une dimension comportant un seul paramètre. Le schéma dimensionnel de notre exemple est représenté selon le formalisme graphique de Pedersen comme suit :

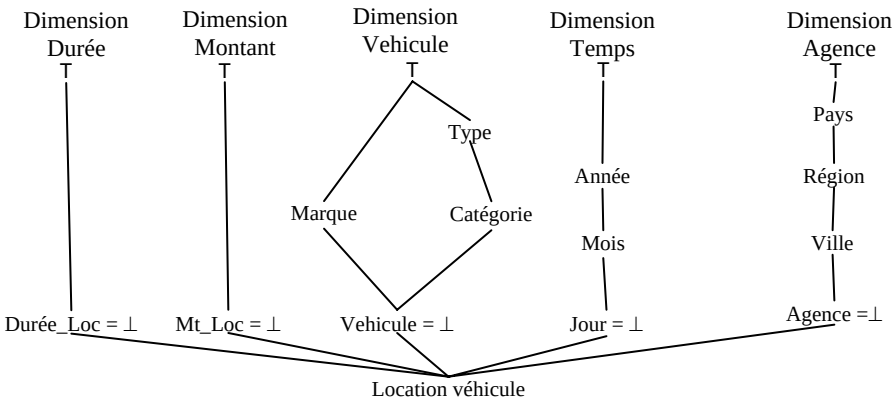


***Figure I.12** : SCHEMA DIMENSIONNEL (PEDERSEN ET AL, 1998)*

Ce modèle ne permet pas la définition des attributs faibles. Les hiérarchies sont dites non strictes, c'est-à-dire qu'elles peuvent ne pas contenir toutes les données de la dimension, mais le modèle ne fournit pas de mécanismes permettant de gérer cette caractéristique et notamment les conflits possibles entre les hiérarchies.

♦ ***YAM² Yet Another Multidimensional Model***

(Abello et al, 2002) propose un modèle conceptuel dimensionnel appelé YAM² basé sur la spécialisation du méta-modèle du diagramme de classe UML.

Le modèle proposé est basé sur la dualité fait – dimension. Une ***dimension*** regroupe les concepts de ***niveau,*** modélisant les paramètres de l'analyse, accompagnés des ***descripteurs*** définissant les attributs faibles. Les dimensions héritent de la classe *Classifier* du méta-modèle d'UML, les niveaux héritent de la classe *Class* et les descripteurs héritent de la classe *Attribute*.

Les sujets d'analyse sont organisés en deux niveaux ; un niveau ***fait*** qui décrit un sujet global partageant les mêmes dimensions et un niveau ***cellule*** qui décompose le fait en sous sujets d'analyse. Chaque cellule englobe un ensemble de mesures. Dans notre exemple, nous présentons un fait *Location véhicule* composé d'une seule cellule *Location* et comportant les mesures *Durée_Loc* et *Mt_Loc*. Comme les dimensions, les faits héritent de la classe *Classifier* d'UML, les cellules de la classe *Class* et les mesures de la classe *Attribute*.

Le modèle proposé est dit "multi étoiles" puisqu'il permet la définition de plusieurs faits et de leurs dimensions. L'utilisation des différents mécanismes orientés objet tels que la généralisation, l'agrégation et la dérivation sont étudiés dans ce modèle dimensionnel.

Exemple 7

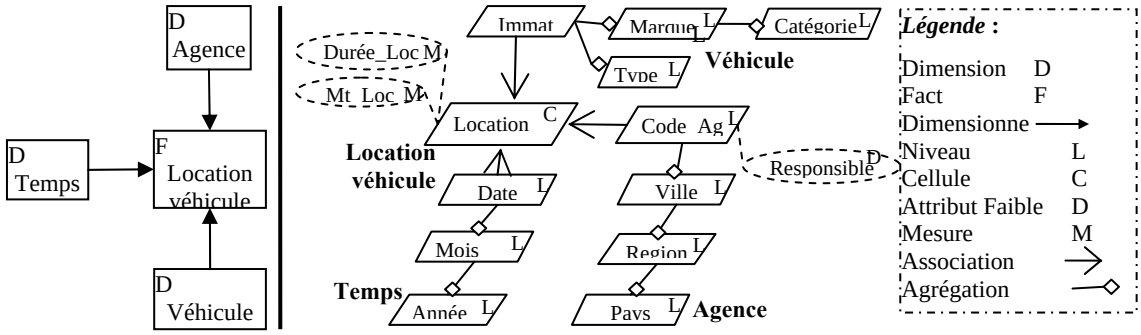


***Figure I.13** : MODELE YAM² : HAUT, MOYEN ET BAS NIVEAUX DE DETAIL (ABELLO ET AL, 2002)*

Le modèle est basé sur trois niveaux de détails ; le plus haut niveau ne comporte que les faits et les dimensions, les moyens et bas niveaux, rassemblés dans le même schéma dans notre exemple, décrivent les paramètres, les attributs faibles et les mesures (voir Figure I.13).

Un ensemble de contraintes d'intégrité est défini dans YAM² afin de réaliser correctement les fonctions d'agrégation des données. Ces contraintes permettent de mettre en évidence les caractéristiques globales des hiérarchies, telle que la relation de dépendances entre les niveaux, mais n'englobent pas les contraintes sémantiques qui peuvent être exprimées entre les instances des hiérarchies.

♦ ***Bilan des modèles objets***

Les modèles dimensionnels objet proposent d'étendre les concepts objet pour la définition des concepts de fait et de dimension. La plupart de ces modèles intègrent les concepts de spécialisation/généralisation et d'agrégation dans la définition des faits et des dimensions. L'intégration de ces concepts pour modéliser les hiérarchies permet d'enrichir la sémantique du modèle.

Néanmoins, le décideur, qui souhaite trouver un modèle simple à analyser, se trouve obligé de maîtriser l'ensemble des concepts objet afin de définir son application décisionnelle. En outre, ces modèles ne permettent pas d'exprimer l'ensemble des contraintes sémantiques qui existent entre les différentes hiérarchies.

2.1.2. Modèles spécifiques

Les modèles purement dimensionnels se basent sur les concepts de fait et de dimension. La simplicité des notations et des concepts ainsi que le souci de se rapprocher de la vision conceptuelle du décideur est le point fort de ces modèles.

♦ ***Modèle dimensionnel des faits (DF)***

(Golfarelli et al, 1998) présente un modèle de données conceptuel et graphique appelé modèle Dimensionnel des Faits (DF). Il propose une méthode semi-automatique pour la dérivation d'un schéma exprimé à l'aide des notations DF à partir du schéma Entité-Association des sources de données opérationnelles.

Le modèle Dimensionnel des Faits se base sur une structure arborescente (cf. Figure I.14).

Exemple 8

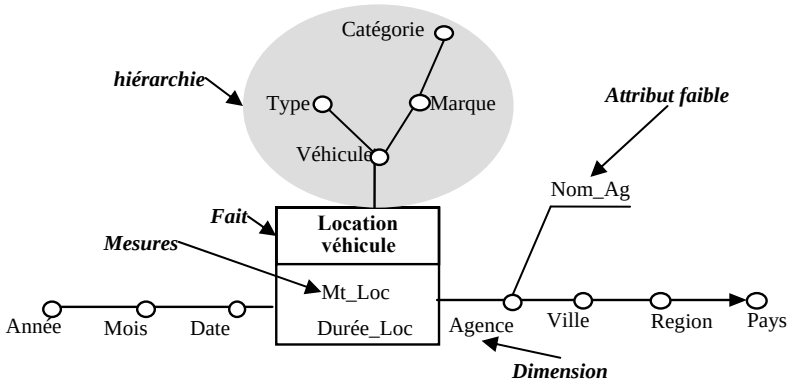


***Figure I.14** : MODELE DIMENSIONNEL DES FAITS (GOLFARELLI ET AL, 1998)*

Le modèle propose un formalisme graphique assez simple à utiliser et à comprendre. Ce formalisme permet de représenter les hiérarchies multiples et les attributs faibles. Néanmoins, le modèle ne fournit pas des noms aux hiérarchies ce qui ne permet pas de les différencier et peut créer des confusions au niveau des parties communes à plusieurs hiérarchies.

♦ ***Object Oriented Multidimensional Data Model (OOMDM)***

(Nguyen et al, 2000) (Bruckner et al, 2001) proposent un modèle dimensionnel spécifique et un méta-modèle objet qui décrit les concepts de ce modèle (diagramme de classes UML).

Le modèle est basé sur les concepts de dimensions, de mesures et de cubes. Une dimension est composée d'un ensemble de paramètres, appelés ***niveaux,*** qui sont organisés selon une fonction d'ordre, notée $\prec_L$, définissant le ***schéma de la dimension***. Les valeurs des paramètres sont appelées ***membres de la dimension*** et sont organisées à leur tour suivant une fonction d'ordre partiel notée $\prec_M$.

Les faits sont définis à l'aide du concept ***cube***. Les mesures sont des entiers ou des réels qui composent les cases du cube. Ces cases sont rassemblées selon des fonctions de regroupement, appelées "Group By", prédéfinis dans le modèle. (Bruckner et al, 2001) étend ce modèle en intégrant la gestion de l'évolution des données dans le temps.

Exemple 9

Ce modèle ne fournit pas de formalisme graphique pour le schéma dimensionnel. La dimension Temps de notre exemple est définie selon ce modèle comme suit :

– *dom* (Temps) = {'all', '1999', 'Mar.1999', '3.Mars.1999', …}. Cet ensemble regroupe toutes les valeurs des paramètres de la dimension *Temps.* Ces valeurs sont organisées suivant la fonction d'ordre $<_M$ comme suit : 'all' $<_M$ '1999', …, 'Mar.1999' $<_M$.'3.Mars.1999'.

– *Levels* (Temps)={All, Année, Mois, Jour} avec dom(*All*)= {'all'}, dom(*Année*)={'1999', …}, dom(Mois)= {'Jan.1999', 'Fev.1999', 'Mar.1999', …}, dom(Jour)= {'1.Jan.1999', '6.Jan.1999', '3.Fev.1999', '3.Mar.1999', …} est l'ensemble des niveaux.

– *DSchema*(Temps) ={All $<_L$ Année, Année $<_L$ Mois, Mois $<_L$ Jour}. La fonction d'ordre $<_L$ permet de définir le schéma de la dimension en fournissant à la fois les paramètres de la dimension et l'ordre de ces paramètres.

Le modèle proposé supporte les hiérarchies multiples mais ne définit pas les attributs faibles permettant de compléter la sémantique des niveaux hiérarchiques. L'inconvénient majeur de cette solution est la non proposition d'un formalisme graphique pour les concepts dimensionnels.

♦ ***Temporal OLAP (TOLAP)***

(Hurtado et al, 1999) propose un modèle dimensionnel qui sépare le contenu de la structure des dimensions. Ce modèle est enrichi, dans (Mendelzon et al, 2003), par des concepts temporels pour supporter l'évolution temporelle des dimensions. Un langage d'interrogation temporel (TOLAP) est associé à ce modèle.

Le concept de base du modèle est *la dimension temporelle* définie par un schéma et un ensemble d'instances. Le schéma est défini par :

– un ensemble fini de niveaux L qui contient le niveau spécifique All représentant le niveau final dans les hiérarchies,

– une fonction d'ordre partiel définie sur les niveaux décrivant les hiérarchies de la dimension.

Ce schéma est représenté par un graphe acyclique dont les nœuds sont les niveaux et les liens sont définis par la fonction d'ordre. Sur chaque lien, un intervalle temporel définit la date de début et la date de fin de validité de ce lien. Un lien qui n'a pas d'intervalle associé est valide tout le temps.

Le deuxième concept est le *fait temporel* défini aussi par un schéma et un ensemble d'instances.

Exemple 10

Nous reprenons l'exemple de la dimension Agence en considérant que le schéma de cette dimension a évolué à partir de l'instant t_1 avec l'ajout du nouveau niveau *Zone*. Ce schéma sera représenté comme suit :

– L = {CodeAg, Ville, Région, Département, Pays, All},

– La relation $\preceq$ contient les données suivantes : CodeAg $\preceq$ Ville, Ville $\preceq$ Région, Région $\preceq$ Pays, Pays $\preceq$ All, CodeAg $\preceq_{t>t1}$ Zone, Zone $\preceq_{t>t1}$ All.

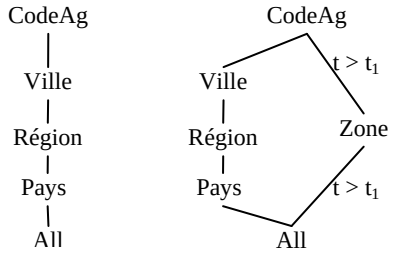


***Figure I.15** : EVOLUTION DU SCHEMA DE LA DIMENSION AGENCE*

Le modèle proposé intègre un ensemble de contraintes (Hurtado et al, 2002) qui visent à résoudre les conflits entre les instances des dimensions en indiquant implicitement l'ensemble des instances appartenant à chaque hiérarchie. La définition de ces contraintes est incomplète puisqu'elles sont définies au niveau d'une seule dimension alors que des conflits peuvent exister entre les instances de dimensions différentes (Ghozzi et al, 2003b).

♦ ***Modèle MD***

(Cabibbo et al, 2000) (Torlone, 2003) proposent un Modèle Dimensionnel (MD) formel basé sur la notion de ***f-table***. Une *f-table* est une fonction reliant une combinaison des valeurs des paramètres aux mesures. Les dimensions sont organisées en niveaux hiérarchiques. Les hiérarchies sont structurées à l'aide d'une fonction d'ordre partiel, notée $\preceq$, définie sur les niveaux hiérarchiques de la dimension. Au niveau de chaque dimension, les valeurs des différents niveaux hiérarchiques sont reliées par une famille de fonctions appelée *Roll-up*. Le schéma dimensionnel est défini par le couple (D, F) avec D un ensemble fini de dimensions et F un ensemble fini de f-tables.

Exemple 11

Selon ce modèle, le schéma dimensionnel S= (D, F) de notre exemple aura la forme suivante :

– D = {Agence, Véhicule, Temps} où la dimension Agence est définie par le triplet <AGENCE, $\leq_{Agence}$, Roll-Up_{Agence}> avec AGENCE = {Code_Ag, Ville, Région, Pays} et Code_Ag $\leq_{Agence}$ Ville $\leq_{Agence}$ Région $\leq_{Agence}$ Pays comme relation d'ordre entre les niveaux. Chaque niveau est associé à un domaine. Par exemple, $Dom_{Région}$= {Est, Ouest, Sud, Nord}. D'où, une fonction de la famille Roll-Up_{Agence} est Roll-$Up^{Région}_{Ville}$ (Lyon)=Centre. Les dimensions *Véhicule* et *Temps* sont définies de la même manière.

– F = {Location} avec Location est une F-table définie par Location [Periode : Date, Véhicule : Immat, Agence : Code_Ag] $\rightarrow$ [Mt_Loc : numéric, Nb_loc : numéric]. Les attributs Période, Véhicule et Agence font partie de la F-table et sont associés à des niveaux hiérarchiques des différentes dimensions. La deuxième partie de la fonction comporte les mesures du fait.

Le modèle permet de décrire les attributs faibles aux différents niveaux. Par exemple, le responsable de l'agence associé au niveau Code_ag est défini par Responsable (Code_Ag) : string

Une représentation graphique de ce modèle est fournie. Dans cette représentation, basée sur un graphe orienté, les faits sont représentés par des nœuds en gras et les dimensions par un sous-graphe encerclé. Chaque sous-graphe de dimension contient un ensemble de nœuds représentant les niveaux hiérarchiques reliés par des arcs représentant la fonction Roll-Up. Un attribut faible est représenté par un nœud en dehors du sous graphe relié au niveau hiérarchique qu'il décrit. Les mesures sont aussi représentées par des nœuds reliés au nœud fait.

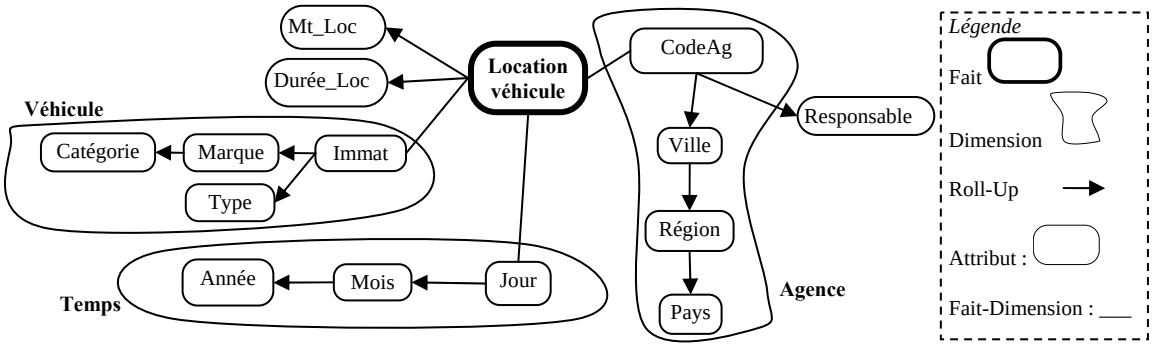


***Figure I.16** : MODELE MD (CABIBBO ET AL, 2000)*

Le modèle permet le traitement symétrique des paramètres et des mesures. Il permet de modéliser des hiérarchies multiples grâce à la fonction d'ordre partiel ≤, mais le formalisme ne fournit pas de nom pour chaque hiérarchie. En outre, le modèle ne fournit pas de support pour les contraintes sémantiques entre les hiérarchies et ne gère pas les conflits éventuels entre les instances des dimensions.

2.1.3. Bilan

Au niveau conceptuel, les travaux sur la modélisation dimensionnelle sont de trois types :

– les travaux qui se basent sur le paradigme Entité–Association,

– les travaux qui se basent sur le paradigme objet,

– les travaux spécifiques qui sont purement dimensionnels basés sur les concepts de fait et de dimension.

Dans (Pedersen et al, 1998) et (Pedersen et al, 2004b), onze critères de comparaison des modèles dimensionnels ont été proposés. Ces critères sont définis dans un contexte d'entrepôt de données médicales pour un modèle basé sur le paradigme objet. D'autres comparaisons ont été faites dans (Chaudhuri et al, 1997), (Blashka et al, 1998), (Vassiliadis et al, 1999).

Afin de pouvoir classifier les travaux réalisés dans ce domaine, nous proposons la liste des critères suivante (qui étend celle définie dans les travaux de comparaison cités) :

– *Définition d'un formalisme graphique*. Plusieurs modèles proposent un formalisme graphique spécifique à la prise de décision. Ces formalismes visent à faciliter la compréhension du modèle par les décideurs.

– *Modélisation des dimensions.* Au niveau des dimensions, nous pouvons distinguer les critères suivants :

  - *Représentation explicite des hiérarchies*. Plusieurs modèles fournissent les opérations qui permettent de capturer les différentes dimensions avec leurs hiérarchies d'une manière explicite (E), d'autres implicitement (I).

  - *Support des hiérarchies multiples.* Les modèles qui intègrent ce critère permettent de spécifier plusieurs hiérarchies de paramètres au sein d'une même dimension.

- *Support des attributs non dimensionnels (faibles).* Certains paramètres dans les hiérarchies nécessitent d'être accompagnés d'informations qui complètent leur sémantique. Ces informations sont modélisées par des attributs faibles situés au même niveau hiérarchique que le paramètre concerné.

– *Modélisation des faits*.

- *Agrégation correcte et significative des mesures.* Le modèle doit fournir pour chaque mesure les différentes fonctions d'agrégation compatibles avec son type permettant d'obtenir un résultat correct et significatif.
- *Faits multiples partageant les dimensions.* Le regroupement de plusieurs sujets d'analyse facilite la corrélation entre ces sujets. Par exemple, la baisse des ventes pour le mois de janvier peut s'expliquer par une baisse des achats ou une rupture de stock. Si l'entrepôt est conçu pour suivre les ventes et les achats, nous pouvons comparer les deux faits et réaliser un rapport global en se basant sur les dimensions partagées *Temps* et *Produit*. On parle alors de forage transversal ou drill across (Abello et al, 2003).

– *Séparation du contenu et de la structure*. Ce critère est relatif à la séparation des instances représentant le contenu du cube du schéma des données dimensionnelles (Gyssen et al, 1997). Les modèles ne respectant pas ce critère se focalisent sur la définition d'opérateurs algébriques pour la manipulation des données mais occultent complètement la définition du schéma dimensionnel (notamment la spécification des hiérarchies de paramètres pour les opérations de forages).

– *Historique des données.* L'introduction de la dimension *Temps* ne permet pas encore d'exprimer des requêtes temporelles complexes comparables aux possibilités offertes dans les bases de données temporelles. Ce critère permet de démarquer les travaux qui ont étudié la gestion du temps dans les modèles dimensionnels (Pedersen et al, 1999) (Mendelzon et al, 2003). Il traite de la conservation de l'historique des mises à jour des données dimensionnelles.

– *Expression des contraintes.* Afin d'éviter toute incohérence, le modèle dimensionnel doit capturer les différentes contraintes sémantiques et structurelles au niveau des hiérarchies (Hurtado et al, 2002). Ce critère garantit la définition d'un modèle dimensionnel fiable qui exprime les contraintes que les données des hiérarchies doivent respecter.

Le tableau suivant présente une étude synthétique des différents modèles dimensionnels de niveau conceptuel.

Critères \ Modèles	StarER	Modèle E/A Dimensionnel	GOLD	EMDM	Yam²	DF	OOMDM	TOLAP	MD
Définition d'un formalisme graphique	√	√	√	√	√	√			√
Représentation explicite des hiérarchies	√		√	√	√	√	√	√	√
Support des hiérarchies multiples.	√	√	√	√	√	√	√	√	√
Expression des contraintes								√	
Support des attributs faibles	√	√	√		√	√			√
Agrégation des mesures	√	√	√	√	√	√	√		
Définition des faits multiples		√	√		√			√	√
Séparation du contenu et de la structure	√	√	√	√	√	√	√	√	√
Historique des données				√			√	√	

***Tableau I.2** : COMPARATIF DES TRAVAUX AU NIVEAU CONCEPTUEL*

Dans le contexte de la modélisation dimensionnelle, la plupart des travaux au niveau conceptuel séparent le contenu de la structure des données et intègrent la représentation explicite des hiérarchies multiples. Plusieurs de ces modèles supportent les attributs faibles permettant d'ajouter de la sémantique aux paramètres. D'autres proposent un formalisme graphique plus ou moins complexe pour les données décisionnelles. Quelques modèles seulement intègrent la définition de faits multiples et expriment les fonctions d'agrégation adaptées aux mesures. Un seul modèle intègre l'expression de contraintes au niveau du modèle dimensionnel (Hurtado et al, 2002).

2.2. Niveau logique

Dans cette section, nous décrivons les travaux qui proposent des modèles dimensionnels basés sur un modèle logique de données. Ainsi, nous retrouvons les modèles ROLAP basés sur un modèle relationnel, OOLAP basés sur un modèle objet et MOLAP basés sur un modèle dimensionnel.

2.2.1. Modèles ROLAP

Les modèles ROLAP reprennent les concepts dimensionnels de base (fait, dimension et hiérarchie) et les transforment en tables relationnelles.

♦ ***Modèle dimensionnel***

(Kimball et al, 2002) est considéré comme une référence dans les bases dimensionnelles, où est défini le célèbre ***schéma en étoile***. Dans ce schéma :

- une dimension est représentée par une table ayant comme clé primaire, l'identifiant de la dimension (paramètre de granularité la plus faible) et comme attributs, les attributs de l'analyse (paramètres et attributs faibles)
- un fait est représenté par une table ayant comme attributs les mesures d'analyse du fait. La clé primaire de cette table est composée de la concaténation des clés des différentes tables de dimension reliées au fait. Des contraintes d'intégrité référentielle sont définies entre les attributs de la clé de la table de fait et les clés des tables de dimension.

Exemple 12

Ce schéma représente la transformation ROLAP du schéma en étoile de la Figure I.3.

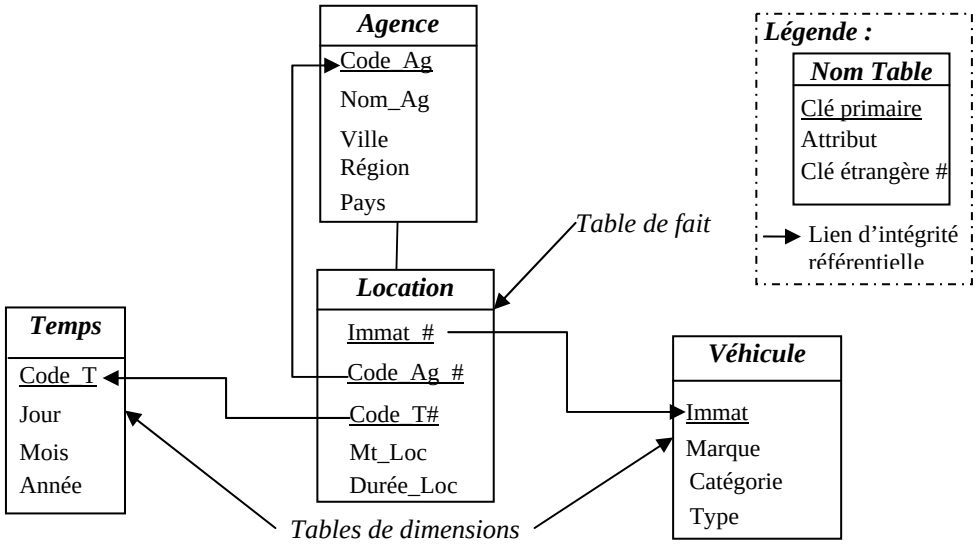


***Figure I.17** : EXEMPLE DE SCHEMA EN ETOILE*

Il est possible de normaliser les tables de dimension pour obtenir le ***schéma en flocon de neige***. Chaque dimension est transformée en plusieurs tables relationnelles en respectant la troisième forme normale. Ainsi, chaque niveau hiérarchique est transformé en une table ayant comme clé, le transformé du paramètre de ce niveau et comme attributs, les attributs faibles qui lui sont attachés. Chaque table de niveau hiérarchique comporte une clé étrangère référençant la table du niveau suivant dans la hiérarchie. Par exemple, la table *Région* est référencée par la table *Ville* et référence la table *Pays*.

Exemple 13

La normalisation des dimensions de l'exemple 12 permet d'obtenir le schéma en flocon de la Figure I.18. Dans ce schéma, nous transformons la dimension *Agence* en quatre tables relationnelles correspondant aux niveaux hiérarchiques *Code_Ag*, *Ville*, *Région* et *Pays*.

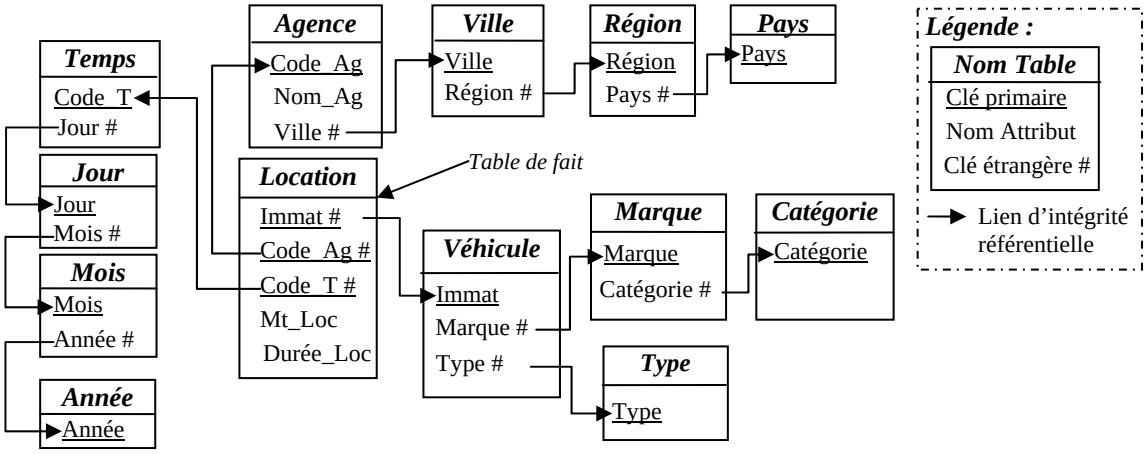


***Figure I.18** : EXEMPLE DE SCHEMA EN FLOCON DE NEIGE*

Le schéma en flocon permet d'expliciter la structure des hiérarchies en normalisant les dimensions. Par contre, il nécessite de définir des jointures entre les différentes tables de niveaux hiérarchiques pour interroger les données d'une dimension. Le temps de

réponse aux requêtes mettant en jeu un volume très important de données devient alors inacceptable dans un contexte d'analyse interactive (Kimball et al, 2002).

Il est possible d'appliquer les mêmes principes pour la transformation d'un schéma en constellation.

♦ ***Cube Dimensionnel***

Dans (Li et al, 1996), un modèle ROLAP est proposé pour les données dimensionnelles. Ce modèle est composé d'un ensemble fini de ***cubes dimensionnels*** et d'un ensemble fini de relations. Chaque ***cube*** est représenté par une mesure et un ensemble de tables de dimensions, tel que à chaque combinaison de n-uplets (un n-uplet de chaque dimension) correspond une valeur de mesure. Les hiérarchies sont exprimées par des relations de regroupement (cf Figure I.19 (b)).

Exemple 14

La Figure I.19 (a) montre le cube dimensionnel Location des Véhicules formé des relations *Véhicule*, *Agence* et *Temps* et de la mesure *Mt_Loc*. La deuxième table de la Figure I.19 (b) présente la relation de regroupement reliant les villes de "Toulouse" et d'"Albi" à la région "Midi-pyrénées" et la ville de "Bordeaux" à la région "Aquitaine".

Véhicule		***Agence***			***Temps***			
Immat	***Marque***	***Code_Ag***	***Ville***	***Région***	***Date***	***Mois***	***Année***	***Mt_Loc***
MY98	Peugeot	AgTLse	Toulouse	Midi-pyrénées	12/07/04	Juillet04	2004	200
ZM76	Citroën	AgBor	Bordeaux	Aquitaine	15/06/04	Juin 04	2004	300

(a) Cube dimensionnel

Région	***Ville***
Midi-pyrénées	Toulouse
Midi-pyrénées	Albi
Aquitaine	Bordeaux

(b) Relation de regroupement

Figure I.19 *: CUBE DIMENSIONNEL (LI ET AL, 1996)*

La définition des hiérarchies est réalisée d'une manière implicite ; le modèle ne fournit pas une structure pour ces hiérarchies, elles sont définies dynamiquement à l'aide de fonctions de regroupement qui sont calculées lors de l'interrogation. En outre, les dimensions ne sont pas partagées entre les cubes, ce qui ne facilite pas la corrélation des analyses.

♦ ***Table N-dimensionnelle***

(Gyssen et al, 1997) propose un modèle dimensionnel en étoile qui distingue la structure du contenu des données. La structure des données est représentée par la notion de ***table N-dimensionnelle*** (Figure I.20 (a)) et les instances par un ensemble de tables relationnelles suivant un schéma en étoile (Figure I.20 (b)). Les hiérarchies ne sont pas explicitées dans le modèle ; elles sont calculées à travers des fonctions.

Exemple 15

Selon le modèle de (Gyssen et al, 1997), à partir de notre exemple d'analyse des locations en fonction du temps et des agences, nous obtenons une table N-dimensionnelle représentée par la Figure I.20 ci-dessous.

LOCATION			TEMPS				
			Années	2002			
			Mois	Janv	Fev	...	Déc
AGENCES	Régions	Ville	(Mt_Loc, Durée_Loc)				
	MP	Toulouse		(80, 8)	(120, 12)		(100, 8)
		Albi		(120, 15)	(100, 10)		(170, 18)
	Languedoc Rousillon	Montpellier		(100, 7)	(50, 5)		(250, 35)
		Nimes		(220, 40)	(130, 20)		(240, 30)

(a) Table N-dimensionnelle

rAgences

Tid	*Régions*	*Ville*
A1	MP	Toulouse
A2	MP	Albi
A3	Languedoc Rousillon	Montpellier
A4	Languedoc Rousillon	Nimes
….	…	…

rTemps

Tid	Mois	Année
T1	janvier	2002
T2	février	2002
T3	mars	2002
T4	avril	2002
….	….	….
T12	decembre	2002

rm

Agence.Tid	Temps.Tid	Mt_Loc	Durée_Loc
A1	T1	80	8
A1	T2	120	12
A1	T12	100	8
…	…	…	…
A4	T12	240	30

(a) Instance de la table N-dimensionnelle

***Figure I.20** : TABLE N-DIMENSIONNELLE (GYSSEN ET AL, 1997)*

La simplicité de l'approche des tables N-dimensionnelles facilite l'analyse des données décisionnelles. Le modèle ne permet pas de définir plusieurs faits. Le calcul des hiérarchies multiples à l'aide des fonctions complique les opérations d'interrogation. En outre, le modèle ne permet pas d'exprimer les contraintes sémantiques entre les données des hiérarchies (Hurtado et al, 2002).

2.2.2. Modèles OOLAP

L'OOLAP vise à combiner les avantages des deux approches ROLAP et MOLAP et à limiter leurs inconvénients (Buzydlowski et al, 1998). D'un côté, comme le ROLAP, l'OOLAP se base sur un standard, le paradigme objet, proposé par l'ODMG. De l'autre côté, comme pour le MOLAP, les requêtes objet sont assez flexibles et extensibles pour réaliser facilement les opérations de manipulation OLAP. Nous présentons dans cette section le travail de recherche qui utilise le modèle OOLAP.

♦ ***Object OLAP***

(Buzydlowski et al, 1998) propose un modèle dimensionnel comportant trois catégories d'objets : objets de données, de contrôle et d'interface. Les objets de données sont les faits et les dimensions. Les objets de contrôle sont les requêtes, les opérations OLAP et les classes de manipulation. Enfin, les objets d'interface sont les outils permettant de visualiser les résultats des objets de contrôle.

Dans ce modèle, les dimensions sont modélisées par deux types de classes :

- Toute dimension ne comportant pas d'attributs faibles est transformée en une ***classe non associative.***
- Toute dimension comportant des attributs faibles est transformée en un ensemble de ***classes associatives*** (une classe par niveau hiérarchique).

Ce modèle ne permet de définir qu'un seul fait dans le même schéma dimensionnel. Les hiérarchies sont strictes. La rigidité des hiérarchies ne permet pas d'exprimer l'hétérogénéité des dimensions dont les instances peuvent appartenir à différentes hiérarchies (Hurtado et al, 2002).

2.2.3. Modèles MOLAP

L'approche MOLAP[2] se base sur un modèle dimensionnel structurant les données dans des cubes de données, des matrices ou des vecteurs à n dimensions. Ces structures optimisent les temps d'accès aux données et réduisent les temps de réponse aux requêtes (Gardarin, 1999).

♦ ***Hypercube***

(Agrawal et al, 1997) présente un modèle dimensionnel basé sur le concept **d'hyper cube**. Le modèle organise les données en plusieurs cubes. Un cube est composé de *K* dimensions et un ensemble de valeurs de mesures (case du cube). Une case du cube peut prendre trois valeurs possibles ; (i) un n-uplet comportant la valeur des mesures correspondant à cette combinaison, (ii) la valeur 1 indique que la combinaison existe, (iii) la valeur 0 indique que la combinaison des valeurs des dimensions n'existe pas.

Exemple 16

La Figure I.21 présente l'hypercube des locations de véhicules dont les dimensions sont *Agence*, *Véhicule* et *Temps* et les cases contiennent les montants et la durée de location. Par exemple, la case correspondant aux agences de la ville de "Toulouse", aux véhicules de marque "Citroën" et à l'année "2000" de la dimension *Temps* contient la valeur <220, 8> pour les montants et la durée des locations. Le deuxième cube de la figure comporte les mêmes dimensions mais en remplaçant les cases contenant des n-uplets par la valeur 1, les autres cases vides correspondent à la valeur 0.

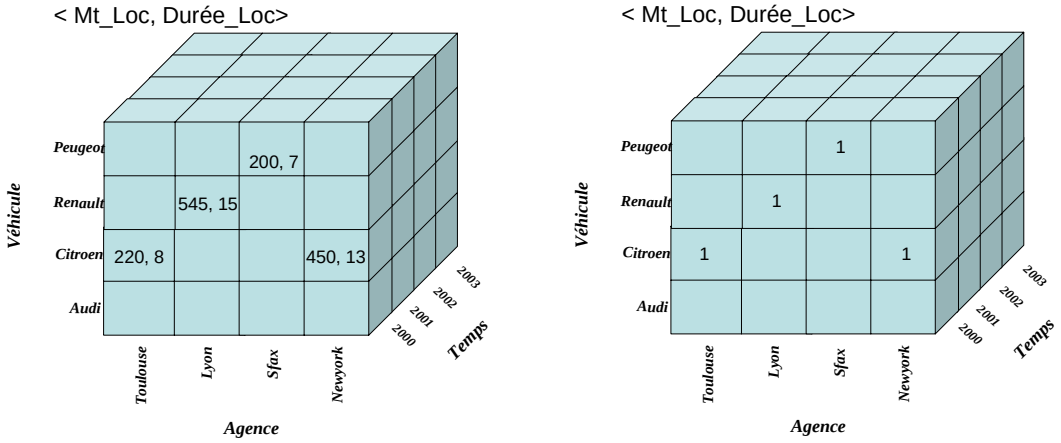


***Figure I.21** : HYPERCUBE DE (AGRAWAL ET AL, 1997)*

Le modèle repose sur le concept d'*hypercube* ; la distinction entre le contenu et la structure dimensionnelle n'est pas explicite dans ce modèle. Les hiérarchies n'ont pas de structures formelles. En effet, le modèle se base sur des fonctions pour la définition des hiérarchies multiples.

♦ ***Cube de base***

(Vassiliadis, 1998) et (Jarke et al, 2003) proposent un modèle MOLAP basé sur la notion de *Cube*.

[2] http://www.01net.com/outils/imprimer.php?article=179791

Pour chaque dimension d'analyse, un ensemble de niveaux hiérarchiques est défini formant un treillis. Un niveau hiérarchique appartient à une seule dimension et il est associé à un domaine de valeurs. Parmi les dimensions, nous retrouvons une dimension spécifique *M* qui représente les mesures de l'analyse.

Un schéma dimensionnel est défini comme un ensemble de dimensions et un ***cube de base***. Un ***cube de base*** est défini par le triplet $<D_b, L_b, R_b>$, tel que :

- $D_b = <D_1, D_2,\ldots, D_n, M>$ est un ensemble de dimensions comportant la dimension spécifique *M* représentant les mesures,
- $L_b = <DL_{b1}, DL_{b2}, \ldots, DL_{bn}, ML>$ est un ensemble de niveaux hiérarchiques comportant le niveau spécifique des mesures *ML* et
- R_b est une relation dont les instances sont de la forme $x = [x_1, x_2, ..., x_n, m]$ avec $x_i \in Dom(DL_i)$ et $m \in dom(ML)$.

A partir de ce cube et en utilisant les opérateurs dimensionnels, d'autres cubes peuvent être obtenus.

Exemple 17

Le cube de base de notre exemple est représenté par la Figure I.22. Dans cette figure, nous retrouvons le cube de base $C_b = < D_b, L_b, R_b >$ avec :

- D_b = <Agence, Temps, Véhicule, Location> l'ensemble des dimensions comportant la dimension spécifique M = Location,
- L_b = <Code_Ag, Immat, Date, Mt_Loc> l'ensemble des niveaux hiérarchiques les plus détaillés et
- R_b, représentée par le tableau ci-dessous, la relation comportant les instances de ce cube.

Agence	Véhicule	Temps	Location
AgTlse	98MY	01-01-2004	200
AgTlse	56MZ	01-01-2004	650
AgTarb	69MZ	01-01-2004	320
AgTarb	63MZ	18-02-2004	500
…..	…	…	…

***Figure I.22** : CUBE DE BASE (VASSILIADIS, 1998)*

Ce modèle ne distingue pas suffisamment le contenu de la structure dimensionnelle ; le cube contient à la fois la structure des dimensions et les données. Dans ce modèle, les faits ne sont pas explicitement définis ; les mesures sont considérées comme des niveaux hiérarchiques d'une dimension.

2.2.4. Bilan

Afin de comparer les travaux au niveau logique, nous reprenons les mêmes critères que ceux du niveau conceptuel. Ces modèles reprennent les concepts du niveau conceptuel (fait, dimensions, hiérarchies) dans un contexte relationnel (ROLAP), objet (OOLAP) ou dimensionnel (MOLAP).

Critères \ Modèles	*Modèle Dimensionne*	*Cube dimensionnel*	*Table N-Dimensionnelle*	*Object OLAP*	*Hypercube*	*Cube de base*
Définition de formalisme graphique						
Représentation explicite des hiérarchies				√		√
Hiérarchies multiples.				√		√
Expression des contraintes						
Support des attributs faibles		√	√		√	
Agrégation des mesures	√					
Définition des faits multiples						
Séparation du contenu et de la structure	√	√	√	√		
Historique des données						

***Tableau I.3** : COMPARATIF DES TRAVAUX AU NIVEAU LOGIQUE*

Représentant les premiers travaux dans le contexte de la modélisation dimensionnelle, les modèles logiques dimensionnels présentent plusieurs lacunes par rapport aux travaux du niveau conceptuel. Le Tableau I.3 montre que la plupart de ces travaux ne distinguent pas la structure des données de leur contenu. Au niveau de la modélisation des dimensions, peu de travaux intègrent la représentation explicite des hiérarchies et la hiérarchisation multiple. Nous remarquons aussi qu'aucun modèle ne propose de formalisme graphique pour les concepts dimensionnels ; nous considérons que le cube est une interface pour la restitution des résultats aux décideurs mais ne présente pas un formalisme graphique pour les concepts dimensionnels. Enfin, les modèles du niveau logique n'intègrent pas l'expression de faits multiples et ne gèrent pas l'historique. Le schéma dimensionnel logique est obtenu à partir d'une transformation du schéma dimensionnel conceptuel. Nous considérons que c'est au niveau conceptuel que les critères définis précédemment doivent être satisfaits.

2.3. Niveau physique

De nombreux travaux dans le domaine des systèmes décisionnels reposent sur les aspects physiques et notamment sur les techniques d'optimisation des processus OLAP. Ces travaux peuvent être classifiés en deux familles selon la technologie appliquée : la matérialisation des vues ou l'optimisation des index. Dans ce qui suit, nous présentons ces deux techniques ainsi que les travaux réalisés dans ce domaine.

2.3.1. Technique de matérialisation des vues

Dans le cadre des systèmes OLAP, la matérialisation des vues est une technique d'optimisation calculant à l'avance les opérations les plus coûteuses en temps d'exécution (agrégation, jointure, ...).

Définition

Une vue matérialisée consiste à stocker dans une base de données le résultat d'une requête calculée à la demande.

Dans le cadre des bases dimensionnelles, (Harinarayan *et al*, 1996) a proposé le concept de ***treillis dimensionnel***. Un treillis présente des vues comportant les mesures de

l'analyse agrégées en fonction des différentes combinaisons des paramètres des dimensions.

La matérialisation des vues de ce treillis permet d'optimiser les temps d'interrogation des données d'analyse. Il existe alors trois possibilités pour sélectionner un ensemble de vues à matérialiser (Ullman, 1996) :

- Matérialiser toutes les vues. Cette approche donne le meilleur temps de réponse pour toutes les requêtes. Mais stocker et maintenir toutes les vues du treillis est impraticable pour une base dimensionnelle importante.
- Ne matérialiser aucune vue. Dans ce cas nous sommes obligés d'accéder aux données détaillées dans les relations de base et les calculs inhérents à l'interrogation sont effectués à chaque requête. Cette solution ne fournit aucun avantage pour les performances des requêtes.
- Matérialiser seulement une partie du treillis. Dans un treillis, les vues sont dépendantes, c'est à dire que la valeur d'une certaine vue peut être calculée à partir d'autres vues. Il est alors souhaitable de matérialiser les vues partagées par plusieurs vues dépendantes. Cette approche a pour but de sélectionner les vues partagées. Cette solution semble la plus intéressante puisqu'elle s'apparente à un compromis entre les deux approches précédentes.

Dans le contexte OLAP, les travaux ont traité deux problèmes majeurs liés aux vues matérialisées :

- le problème de sélection des vues matérialisées,
- le problème de maintenance de ces vues matérialisées.

Nous abordons ces deux problèmes dans les sections suivantes**.**

2.3.1.1. Sélection des vues matérialisées

Plusieurs travaux se focalisent sur le Problème de Sélection des Vues matérialisées (PSV). Les approches proposées dans la littérature visent à déterminer le sous-ensemble de vues dont la matérialisation offre un meilleur compromis entre les temps de réponse aux requêtes et le coût de maintenance des vues matérialisées.

La connaissance préalable ou non des requêtes de manipulation permet de classifier le PSV en deux catégories : le PSV statique et le PSV dynamique.

Le PSV ***statique*** consiste à sélectionner un ensemble de vues à matérialiser afin de minimiser le temps de réponse aux requêtes, le coût de maintenance, ou les deux à la fois, sous la contrainte de la ressource. Le problème suppose donc que l'ensemble des requêtes n'évolue pas. Si des évolutions des requêtes sont enregistrées alors il faut reconstruire les vues à matérialiser.

Le PSV ***dynamique*** propose de combler les lacunes du PSV statique en tenant compte de l'évolution des requêtes analysées. (Kotidis et al, 2001) a proposé un système appelé DynaMat, qui matérialise les vues d'une manière dynamique. DynaMat combine les problèmes de sélection et de maintenance des vues. Ce système enregistre les évolutions des requêtes et matérialise dans chaque cas le meilleur ensemble de vues satisfaisant ces requêtes pour un espace de stockage donné. Pendant les opérations de mise à jour,

DynaMat rafraîchit les vues et si la taille des vues dépasse la capacité de l'espace autorisé, il élimine les vues les moins utilisées.

2.3.1.2. Maintenance des vues matérialisées

Les vues matérialisées sont calculées à partir d'autres vues ou tables de base. Les changements et les mises à jour qui affectent les tables de base doivent être reportées dans les vues matérialisées afin que leurs contenus ne deviennent pas obsolètes. Fondamentalement, ces mises à jour peuvent être réalisées périodiquement, immédiatement à la fin de chaque transaction ou d'une manière différée lors de l'utilisation de la vue par une requête d'un utilisateur. Cependant le simple recalcul du contenu des vues matérialisées s'avère inefficace (très coûteux) dans le cas des bases décisionnelles. Une nouvelle approche basée sur la propagation des mises à jour est alors proposée. Cette approche consiste à détecter les modifications réalisées au niveau des tables de base, à travers un système de journalisation, et à les propager dans les vues, sans recalculer complètement leur contenu. Cette propagation peut se faire d'une manière incrémentale, autonome ou différée.

2.3.1.3. Comparaison des travaux sur la matérialisation des vues

Le Tableau I.4 effectue une comparaison des travaux sur la maintenance et la sélection des vues matérialisées :

- La première colonne classifie les travaux en fonction de la problématique traitée : maintenance (M) ou sélection (S) des vues à matérialiser.
- La deuxième colonne indique le modèle OLAP utilisé. Deux modèles sont utilisés : le ROLAP (*R*) basé sur un modèle relationnel et le MOLAP (*M*) utilisant un modèle dimensionnel spécifique aux applications décisionnelles.
- La troisième colonne décrit le type des algorithmes. Les algorithmes sont de deux types, statiques (S), basés sur un ensemble de requêtes et de contraintes figées, ou dynamiques (D) qui évoluent en fonction du changement des contraintes.
- La quatrième colonne décrit les caractéristiques des vues utilisées tels que :
  - Le type des vues utilisées :
    - virtuelles, notées *V* (les données de la vue restent physiquement stockées au niveau des sources et la vue est calculée au moment de l'interrogation),
    - matérialisées, notées *M* (la vue est stockée physiquement dans l'entrepôt),
    - auxiliaires, notées *A* (une vue auxiliaire est une vue, généralement matérialisée, non directement définie par l'administrateur ; elle est utilisée par le système pour améliorer le fonctionnement de l'entrepôt en conservant des informations supplémentaires).
  - La technique de définition des vues qui peut être effectuée au travers d'opérateurs de sélection, de projection et de jointure (SPJ) ou d'opérateurs de groupement associés à des agrégations (G/A).
- La cinquième colonne décrit les techniques d'optimisation adoptées. Une première caractéristique concerne l'utilisation de treillis dimensionnels (cube de données) ou

d'arbres algébriques pour le calcul des vues. Une deuxième colonne décrit le type d'algorithme utilisé : glouton ou heuristique. Les algorithmes gloutons construisent une solution de façon incrémentale, en choisissant à chaque étape la direction qui est la plus prometteuse. Ce choix localement optimal n'a aucune raison de conduire à une solution globalement optimale. Les algorithmes heuristiques se basent sur des choix d'opérations aléatoires. Deux exécutions du même algorithme sur les mêmes données ne fournissent pas le même résultat. Enfin, nous présentons la nature des contraintes appliquées dans l'algorithme de résolution du PSV (espace stockage (S), temps de maintenance (T)).

Critères / Travaux	Maintenance / Sélection	Modèles de données	Dynamisme	Vues		*Techniques utilisées*		
				Types	Technique de définition	*Schéma de base*	*Algorithme*	*Contraintes*
(Baralis et al, 1997) (Paraboschi et al, 2003)	**S**	**M**	**S**	**M**	**SPJ, A/G**	Treillis	Heuristique	
(Gupta et al, 1997, 1999)	**M, S**	**R**	**S**	**M**	**SPJ**	Arbre Alg	Glouton, A*	T
(Harinarayan et al, 1996)	**S**	**M**	**S**	**M**	**SPJ, A/G**	Treillis	Glouton	S
(Kotidis et al, 2001)	**M, S**	**R**	**D**	**M**	**SPJ, A/G**	Treillis	Heuristique	S, T
(Theodoratos et al, 1999)	**S**	**R**	**S**	**M**	**SPJ, A/G**	Arbre Alg	Heuristique	T
(Yang et al, 2000)	**M, S**	**R**	**S**	**M**	**SPJ, A/G**	Arbre Alg	Heuristique	

***Tableau I.4** : CARACTERISTIQUES DES TRAVAUX SUR LA MATERIALISATION DES VUES*

Nous constatons que la plupart des algorithmes de sélection des vues n'exploitent pas la structure hiérarchique dans l'optimisation de la sélection. Or, la sémantique des hiérarchies peut simplifier le processus de sélection.

2.3.2. Optimisation des index

Les index sont des structures physiques permettant un accès direct aux données. Ils jouent un rôle particulièrement prépondérant dans les bases de données en général et en particulier, dans les systèmes d'aide à la décision dans le sens où ils réduisent le coût des réponses à des requêtes très souvent complexes. Ces travaux n'entrent pas dans notre cadre d'étude mais présentent un important axe de recherche. Nous nous limitons dans cette section à présenter les principales techniques utilisées en optimisation d'index dans le cadre des bases de données dimensionnelles.

2.3.2.1. Index binaires

Les index binaires (Bitmap) sont très utilisés dans ces systèmes qui gèrent un volume important de données et de requêtes ad hoc. L'emploi de ces index est recommandé pour des attributs à faible cardinalité. Dernièrement, (Wu et al, 2004) et (Lim et al, 2004) ont proposé de nouveaux algorithmes adaptant ces index aux attributs de grande cardinalité. Cette indexation consiste à créer un vecteur de bit pour chaque valeur de la colonne indexée. Si on prend l'exemple d'une relation *Personne* comportant la colonne *Sexe*, deux vecteurs binaires "homme" et "femme" peuvent être créés. Chaque vecteur est composé d'un ensemble de bits représentant les différentes lignes de la table analysée. Chaque bit dans le vecteur "homme" est égal à 1 si la valeur de la ligne est "homme" et 0 sinon. Ce

type d'index facilite les opérations de regroupement et de jointure souvent réalisées dans les bases dimensionnelles. Néanmoins, le coût de maintenance de ces index peut être élevé car ils doivent être actualisés à chaque nouvelle insertion d'un n-uplet.

2.3.2.2. Index de jointure

Des index spécialisés, appelés ***index de jointure***, ont été proposés pour accélérer les opérations de jointure dans les bases transactionnelles (Valduriez, 1987). Un index de jointure est une table à deux colonnes, contenant les identifiants des n-uplets de deux tables jointes. Ce genre d'index est souhaité pour les requêtes des systèmes OLTP car elles utilisent souvent des jointures entre deux tables. Par contre, les requêtes décisionnelles définies sur un schéma en étoile possèdent plusieurs jointures (entre la table des faits et plusieurs tables de dimensions). Pour résoudre ce problème, (Red, 1997) a proposé un nouvel index appelé ***index de jointure en étoile*** (star join index), adapté aux requêtes définies sur un schéma en étoile. Un index de jointure en étoile peut concerner n'importe quelle combinaison contenant la clé de la table de fait et une ou plusieurs clés primaires des tables de dimension. Ce type d'index est dit complet s'il est construit en joignant toutes les tables de dimension avec la table de fait. Un index de jointure partiel est construit en joignant certaines des tables de dimension avec la table de fait. En conséquence, l'index complet est bénéfique pour n'importe quelle requête posée sur le schéma en étoile. Il exige cependant beaucoup d'espace pour son stockage.

2.4. Synthèse des modèles dimensionnels

Nous avons présenté dans cette section les différents travaux dans le contexte de la modélisation dimensionnelle, classifiés en trois niveaux d'abstraction : conceptuel, logique et physique. Ces travaux visent à organiser les données d'une manière adaptée aux analyses décisionnelles et à optimiser l'accès et l'interrogation de ces données.

Nous avons réalisé un comparatif de ces travaux en se basant sur différents critères assurant la bonne qualité et l'efficacité de la modélisation dimensionnelle. Plusieurs modèles intègrent la plupart de ces critères, tels que les hiérarchies multiples, la représentation explicite de ces hiérarchies, la séparation entre la structure et le contenu et la définition de faits multiples. Par contre, peu de travaux intègrent la définition de contraintes dans le modèle dimensionnel (Carpani et al, 2001) (Hurtado et al, 2002). Or, la définition des contraintes et des règles de gestion permet d'améliorer la qualité de la conception, l'implantation et la maintenance de la base dimensionnelle (Samtani et al, 1998). Une étude des différents types de contraintes qui peuvent exister dans le modèle dimensionnel s'avère nécessaire afin de permettre leur intégration dans ce modèle.

Nous proposons dans la section suivante d'étudier les contraintes dans le cadre des bases de données d'une manière générale. Puis nous présentons un état de l'art des modèles dimensionnels qui intègrent partiellement ces contraintes.

3. Expression des contraintes : Etat de l'art

Assurer l'intégrité des données dans un système d'information transactionnel est un souci permanent sur lequel repose la fiabilité de tout le système (Codd, 1970). Ce besoin de fiabilité est encore plus important dans les systèmes décisionnels où les décisions

reposent sur le résultat de l'interrogation des données gérées par ce système (Codd et al, 1993).

3.1. Contraintes et bases de données

Dans le domaine des bases de données, l'intégrité recouvre plusieurs notions (Ferrat, 1983). D'une part, ces notions couvrent la protection technique de la base tels que le contrôle de concurrence, la protection des données et la sécurité des données de manière à maintenir la cohérence de la base en cas d'incidents techniques (mécanismes de reprise). D'autre part, les notions d'intégrité concernent la protection de la sémantique, notamment :

- l'intégrité de la démarche de conception (Lapujade, 1997) assurant la validation des étapes de conception du schéma de données,
- l'intégrité du modèle visant à conserver la qualité de l'information de la base en validant à tout moment la sémantique des structures et des valeurs des données.

Ces contraintes d'intégrité sont peu étudiées dans les modèles dimensionnels dédiés à l'aide à la prise de décision (Hurtado et al, 2002) (Hümmer et al, 2002). Afin de pouvoir exprimer les contraintes dans un modèle dimensionnel, nous avons étudié les typologies de contraintes proposées dans le domaine des bases de données.

Les contraintes analysées peuvent être classifiées en plusieurs catégories en fonction des critères de classification utilisées (Ferrat, 1983) (Lapujade, 1997). Ainsi, nous retrouvons les contraintes *statiques* ou *dynamiques* appliquées sur des entités ou sur des transitions d'états, les contraintes *globales* ou *locales* faisant référence au champ d'application de la contrainte, les contraintes *individuelles* ou *ensemblistes* intervenant au niveau d'une ou d'un ensemble d'instances et les contraintes *intra* et *inter concepts* impliquant un seul concept du modèle ou bien plusieurs concepts.

Une autre classification est présentée par (Harkins, 2003). Cette classification décrit les contraintes gérées par les bases de données relationnelles et présente trois types d'intégrité :

- l'intégrité d'entité, où chaque enregistrement est identifié de manière unique,
- l'intégrité référentielle, où chaque valeur de clé étrangère doit avoir une valeur de clé primaire correspondante dans une table associée (ou avoir la valeur nulle),
- l'intégrité métier, il s'agit de règles spécifiques de l'entreprise sans aucun lien avec la théorie des bases de données relationnelles.

3.2. Contraintes et modèles dimensionnels

Notre objectif est de modéliser les contraintes d'intégrité au niveau du modèle dimensionnel ainsi qu'au niveau de la démarche de conception d'un schéma dimensionnel. Dans ce cadre, nous proposons la typologie des contraintes suivante :

- **les contraintes liées au modèle** qui permettent une définition cohérente des données et de leurs structures et aident ainsi à une meilleure prise de décision ;
- **les contraintes liées à la démarche** qui agissent sur la démarche de construction et d'alimentation de la Base de Données diMensionnelles (BDM). Ces contraintes décrivent les étapes et leur ordonnancement (Lapujade, 1997).

3.2.1. Les contraintes liées au modèle

Au niveau de la conception des données, le simple énoncé de la structure des données (schéma) ne suffit pas à garantir la pertinence de l'information. Pour aider au maintien de la cohérence des données, il faut spécifier des règles édictant des propriétés que doivent respecter les données de l'application, ce sont les règles d'intégrité (ou contraintes d'intégrité).

Dans le modèle dimensionnel, deux types de contraintes peuvent être recensés. Les contraintes reliées directement à la définition des structures du modèle et les contraintes qui sont basées sur la sémantique de l'application analysée. Nous proposons la typologie suivante : contraintes structurelles et contraintes sémantiques. Au niveau des contraintes structurelles, nous retrouvons les contraintes orientées valeurs ou instances et les contraintes orientées concepts. Au niveau des contraintes sémantiques, nous définissons les contraintes intra et inter concepts.

L'arborescence des contraintes est définie comme suit :

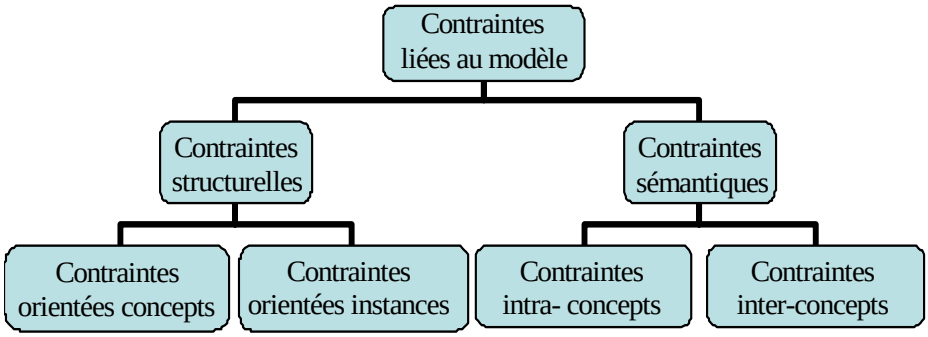


***Figure I.23** : TYPOLOGIE DES CONTRAINTES*

3.2.2. Les contraintes liées à la démarche

Les contraintes liées à la démarche sont peu traitées dans la littérature. En effet, peu de chercheurs proposent une démarche complète de conception des modèles dimensionnels (Golfarelli et al, 1998) (Tryfona et al, 1999) (Lujan et al, 2004).

Deux types de démarches sont définis : la démarche « du bas vers le haut » (Button-Up) et « du haut vers le bas » (Top-down). La première démarche part du modèle des sources opérationnelles pour construire la BDM en se basant sur l'analyse des données pertinentes à la prise de décision. La deuxième démarche (Top-down) part des spécifications des besoins définies par les utilisateurs finaux pour concevoir le modèle multidimensionnel et réalise ensuite la correspondance avec les bases transactionnelles.

3.2.3. Bilan

La plupart des modèles dimensionnels qui ont intégré les contraintes s'intéressent aux contraintes structurelles telles que les contraintes de partition, de connectivité et de disjonction. Ces modèles traitent le problème de l'additivité des mesures dans des hiérarchies hétérogènes (Lehner, 1998) (Hurtado et al, 2002) (Lechtenbörger et al, 2003).

Parmi les rares modèles qui intègrent les contraintes sémantiques, nous citons celui de (Hurtado et al, 2002). Ce modèle intègre les contraintes au niveau d'une même dimension. Il définit des contraintes sur des portions de chemin dans le graphe des

hiérarchies d'une dimension. Ces contraintes permettent d'indiquer si le chemin suivi est valide ou non. Les auteurs proposent aussi un algorithme de validation de ces contraintes.

Un deuxième modèle dimensionnel (Carpani et al, 2001) traite de l'intégration des contraintes sémantiques au niveau conceptuel. Les contraintes définies dans ce modèle sont appliquées au niveau de plusieurs dimensions, d'une seule dimension ou d'un seul niveau hiérarchique. Le modèle propose un langage générique pour exprimer ces contraintes au niveau conceptuel. Les contraintes sont accompagnées d'un formalisme graphique. Les auteurs ne présentent pas l'implication de ces contraintes au niveau de la manipulation et de la construction des données dimensionnelles.

Parmi les travaux qui proposent une démarche, nous citons ceux de (Golfarelli et al, 1998) qui propose un modèle conceptuel dimensionnel appelé DF avec une démarche de conception du schéma dimensionnel. (Tryfona et al, 1999) propose de son côté une extension du modèle Entité-Association pour supporter l'analyse dimensionnelle et une démarche d'extraction de son modèle appelé Star-ER à partir d'un modèle Entité-Association. (Trujillo et al, 2001) propose un modèle dimensionnel orienté objet, accompagné d'une démarche de conception de l'entrepôt de données à partir des spécifications des utilisateurs finaux validées par un outil d'aide à la conception GOLD (Trujillo et al, 2002) (Lujan et al, 2004).

Nous résumons les contraintes définies dans les modèles dimensionnels dans le tableau suivant :

Contraintes \ Travaux			*(Golfarelli 98)*	*(Tryfona 99)*	*(Trujillo 02)*	*(Hurtabo 01, 02)*	*(Lehner 98)*	*(Lechtenborger 03)*	*(Carpani 01)*
Contraintes liées au modèle	***Structurelles***	***Liées aux concepts***	√	√	√	√	√	√	√
		Liées aux instances			√	√		√	√
	Sémantiques	***Inter concepts***						**P**	
		Intra concepts			√	√		√	**P**
Contraintes liées à la démarche		***Top-down***			√				
		Button-up	√	√			√		

Légende : √: Concept défini dans le modèle, P : concept partiellement défini

***Tableau I.5** : EUDE COMPARATIVE DES MODELES DIMENSIONNELS EXPRIMANT DES CONTRAINTES*

L'expression des contraintes dans le modèle dimensionnel est une problématique qui a été peu traitée par les travaux de recherche (Samtani et al, 1998) (Hurtado et al, 2002). Ces travaux ont proposé d'abord d'intégrer des contraintes structurelles notamment au niveau des hiérarchies **(**Lehner et al, 1998) (Golfarelli et al, 1998) (Tryfona, 1999). Ces contraintes permettent d'assurer une structure valide favorisant l'agrégation des données d'un niveau hiérarchique à un autre. Dans une deuxième étape, d'autres travaux ont intégré les contraintes sémantiques en réalisant que d'autres contraintes relatives au contexte de l'analyse peuvent exister. Néanmoins, ces contraintes n'ont été exprimées qu'au niveau d'un seul concept telles que les contraintes sur les instances des hiérarchies exprimées par (Hurtado et al, 2002).

Nous avons distingué une autre catégorie de contraintes : les contraintes relatives à la démarche de conception. A ce niveau, nous avons considéré les travaux qui proposent une démarche ou un processus de conception des schémas dimensionnels intégrant des contraintes d'ordonnancement sur un ensemble d'étapes ou phases de conception. Ces démarches sont proposées, généralement, dans le cadre d'une méthode de conception. Ainsi, nous proposons dans la section suivante d'étudier les différentes méthodes de conception de schémas dimensionnels.

4. Méthodes de conception dimensionnelle : Etat de l'art

Nous proposons de classer ces méthodes en trois catégories : les méthodes ascendantes, les méthodes descendantes et les méthodes mixtes qui combinent les deux premières.

Les méthodes ascendantes utilisent les sources de données pour définir les besoins des décideurs et pour concevoir les schémas dimensionnels. Ces méthodes considèrent que les informations pertinentes pour la prise de décision se trouvent dans la source (List et al, 2002). Les auteurs de (Moody et al, 2000) indiquent que les besoins des utilisateurs sont très évolutifs et difficiles à définir d'où le risque d'obtenir un schéma instable en se basant sur ces besoins dans la modélisation dimensionnelle.

Par opposition, dans les méthodes descendantes, les données des sources ne sont pas prises en compte car ces méthodes considèrent que l'objectif d'un modèle dimensionnel est de répondre aux besoins des utilisateurs (Kimball et al, 2002). Elles se basent uniquement sur la spécification de ces besoins pour définir les sujets et les axes de l'analyse en négligeant la structure et le contenu des sources à partir desquelles les données décisionnelles sont extraites.

Les méthodes mixtes combinent les deux méthodes précédentes et essayent de combler les lacunes de chacune d'elles. Ces méthodes se basent sur les données sources pour définir le schéma dimensionnel en y intégrant les besoins des utilisateurs (Trujillo et al, 2003).

Dans cette section, nous présentons quelques travaux de recherches réalisés dans le cadre de la modélisation dimensionnelle organisés en trois catégories : descendante, ascendante et mixte. Nous présentons ensuite un bilan qui résume et compare ces travaux.

4.1. Méthodes descendantes

Dans (Kimball et al, 2002) différentes études de cas de bases dimensionnelles sont proposées. La modélisation dimensionnelle est basée sur le schéma en étoile et ses différentes variations (schéma en flocon et schéma en constellation). En outre, une méthode, appelée architecture en matrice de BUS, est proposée pour la construction d'un schéma dimensionnel à partir de la définition des besoins des utilisateurs (Kimball et al, 2002). Cette méthode permet de collecter les différents sujets d'intérêt dans l'entreprise et de les combiner avec les différents axes d'analyse pour former une matrice. Cette matrice globale englobe plusieurs sujets d'analyse et l'ensemble de leurs dimensions. Ces travaux ne proposent pas de méthode formelle de conception et de construction d'une base dimensionnelle. Ainsi, nous ne trouvons ni une démarche formelle de spécification des besoins décideurs, ni un outil de transformation de ces besoins dans la matrice proposée.

De plus, nous remarquons que, dans ces travaux, Kimball ne traite pas la modélisation conceptuelle. Ces travaux, basés exclusivement sur une approche ROLAP, se situent au niveau logique.

(Tsois et al, 2001) propose une méthode, permettant de construire le schéma dimensionnel à partir des besoins décideurs, basée sur un modèle conceptuel des données dimensionnelles, appelé MAC. Les auteurs dressent une liste de requêtes définies en fonction des besoins des utilisateurs. Ils définissent ensuite un ensemble minimal de concepts (fait, dimension, cube) permettant de modéliser correctement ces besoins. Ainsi, le schéma construit décrit les besoins et répond aux requêtes déjà définies. Cette description des besoins est réalisée à l'aide d'un ensemble de tables. Les colonnes de ces tables sont composées d'un ensemble de paramètres des dimensions et d'une ou plusieurs mesures. La méthode proposée fournit un schéma conceptuel partiel qui ne décrit que les hiérarchies de dimensions. Ainsi, nous ne retrouvons pas un formalisme qui décrit le modèle MAC en totalité.

(Prat et al, 2002) propose une méthode de développement d'une base dimensionnelle basée sur les notations UML. Cette méthode adopte l'organisation en trois niveaux d'abstraction avec un niveau intermédiaire facilitant le passage entre le diagramme de classes UML et le schéma dimensionnel. Au niveau conceptuel, le concepteur part d'une spécification informelle des besoins des utilisateurs pour créer un diagramme de classes UML. Ce diagramme est enrichi afin d'obtenir un schéma pivot facilitant le passage des concepts UML vers les concepts dimensionnels. Il est obtenu suite à deux étapes de simplification :

- une première étape qui permet de remplacer les méthodes des classes par des attributs étant donné que le modèle dimensionnel ne décrit pas le comportement des faits et des dimensions, et
- une deuxième étape qui transforme les liens multivalués (de type N : M) en liens monovalués (1 : N) tels que les liens entre deux niveaux hiérarchiques et le lien entre le fait et les dimensions.

La transformation du modèle pivot en un modèle dimensionnel est réalisée au niveau logique. Enfin le modèle dimensionnel est implanté en une base de données selon un modèle ROLAP ou MOLAP suivant la plate forme choisie.

4.2. Méthodes ascendantes

(Golfarelli et al, 1998) propose le modèle dimensionnel des faits et une méthode semi-automatique de conception du schéma dimensionnel à partir d'un schéma Entité-Association décrivant les sources. Cette méthode se base sur le schéma logique des données décrivant les sources opérationnelles car souvent la documentation des schémas Entité-Association est incomplète (Golfarelli et al, 2002). Cette méthode est basée sur les trois niveaux d'abstraction : conceptuel, logique et physique. Au niveau conceptuel, la méthode propose un ensemble d'étapes pour la définition des faits, des dimensions et des hiérarchies à partir du schéma de la source. Le fait représente les événements fréquents dans le monde de l'entreprise. Les dimensions sont formées à partir d'une portion du schéma source qui dépend de la relation représentant le fait. Les attributs de cette portion sont extraits puis réorganisés et épurés pour former les hiérarchies. Un schéma dimensionnel, appelé DF, est obtenu à la fin de ces étapes. Ce schéma est par la suite

transformé en un schéma logique relationnel (ROLAP). Au niveau physique, la méthode propose un ensemble d'optimisations possibles basées sur l'analyse des index et des vues matérialisées. Les auteurs présentent un outil d'aide à la conception, appelé WAND, qui se base sur cette méthode. Dans cette méthode, les besoins des utilisateurs ne sont pas définis lors de la conception. L'outil génère, à partir du modèle dimensionnel obtenu, un ensemble de requêtes possibles qui seront par la suite testées par les utilisateurs. Ainsi, l'utilisateur peut exprimer son niveau de satisfaction en visualisant les réponses possibles, fournies par l'outil, à ces requêtes. Le schéma dimensionnel est alors modifié pour tenir compte des remarques des utilisateurs.

(Cabibbo et al, 2000) propose une méthode de conception permettant de définir un schéma logique dimensionnel à partir des sources opérationnelles décrites par un schéma Entité-Association. Cette méthode comporte quatre étapes :

- l'identification des faits et des dimensions : cette identification est réalisée par le concepteur qui analysera le schéma global des sources et détectera les sujets d'intérêt et les dimensions d'analyse ;
- la restructuration du schéma Entité-Association de façon à mettre en relief les faits et les dimensions. Durant cette étape, les faits ayant pour origine des associations, des entités ou des attributs sont restructurés sous forme d'entités. De nouvelles dimensions peuvent être rattachées aux faits, telle que la dimension temps, et les niveaux hiérarchiques dans chaque dimension sont raffinés ;
- la dérivation du graphe dimensionnel à partir du schéma Entité–Association restructuré. Dans cette étape, les auteurs gardent les notations du modèle Entité–Association avec une légère modification qui consiste à mettre en gras les entités représentant les faits et à encercler les sous graphes représentant les dimensions ;
- la translation en un modèle dimensionnel (MD) pour obtenir un schéma dimensionnel ayant des notations spécifiques (cf. Figure I.16).

Cette méthode ne fournit pas d'outils permettant de collecter et de spécifier les besoins des utilisateurs. Ainsi, le schéma dimensionnel obtenu à partir des sources n'est pas confronté aux utilisateurs finaux pour tester leur niveau de satisfaction.

(Moody et al, 2000) propose une méthode de construction d'un schéma dimensionnel à partir des sources opérationnelles décrites par un schéma Entité–Association. Cette méthode offre un ensemble de schémas pour la conception du modèle dimensionnel, tels que les schémas en étoile, en flocon et en constellation. Ces schémas sont tous basés sur des tables relationnelles représentant les faits et les dimensions. Le choix d'un schéma est basé sur le besoin d'expliciter les différentes hiérarchies ou niveaux d'agrégation. Contrairement aux propositions précédentes, cette méthode ne définit pas ses propres notations graphiques, elle utilise les notations du modèle Entité–Association. Les auteurs mettent en avant les avantages d'une méthode ascendante en insistant sur les problèmes rencontrés en adoptant la méthode de Kimball :

- les besoins des utilisateurs sont imprévisibles et sont sujets à plusieurs modifications dans le temps (ceci ne permet pas d'avoir une base solide pour l'analyse) ;
- dans une méthode descendante, la conception peut être erronée si le concepteur ne comprend pas la sémantique des relations entre les données sources ;

– un risque de perte d'informations existe en réalisant des agrégations prématurées basées sur les besoins des utilisateurs.

4.3. Méthodes mixtes

Dernièrement, les méthodes mixtes ont été proposées par plusieurs auteurs. Ces méthodes se basent sur les besoins des décideurs et les données des sources opérationnelles dans la conception du modèle dimensionnel afin de combiner les avantages des méthodes ascendantes et descendantes.

(Carneiro et al, 2002) propose X_META, une méthode de développement des bases dimensionnelles basée sur la démarche en spirale. Cette méthode est composée de plusieurs phases. Une première phase d'introduction permet de concevoir un premier prototype sous forme d'un ensemble de schémas en étoile permettant de répondre à un ensemble de requêtes utilisateurs. A partir de ce prototype, le comité du projet décide de concevoir la base dimensionnelle ou d'arrêter la démarche. Suite à une décision positive, une phase de développement réalise la construction effective du projet. Une dernière phase réalise la production et la maintenance des bases dimensionnelles. Chacune de ces phases est décomposée en sous phases produisant des modules. Par exemple, la phase de développement comporte les sous phases de construction des magasins et de l'entrepôt, de modélisation des méta données et de préparation du référentiel de méta-données.

Les auteurs insistent sur le besoin d'intégrer les décideurs dans le processus de développement du schéma dimensionnel en se basant sur les données collectées à partir des sources opérationnelles. Cette méthode, traitant de la gestion du projet, reste théorique pour la plupart de ses phases. En effet, on ne retrouve pas de modèle, ni d'outils pratiques pour la modélisation des données dimensionnelles.

(Cavero et al, 2001) propose une méthode de conception de base de données dimensionnelles appelée MIDEA. Cette méthode est basée sur un modèle dimensionnel, IDEA (Sanchez et al, 1999) et propose un ensemble d'étapes pour le développement conceptuel, logique et physique des bases dimensionnelles. La première étape, appelé Analyse du Système d'Information (ASI), vise à spécifier les besoins décisionnels en se basant sur la documentation des bases opérationnelles et sur les besoins des décideurs. La deuxième étape réalise la conception du système d'information dimensionnel, notamment, la conception de schémas dimensionnels ROLAP ou MOLAP et leur transformation en modèles physiques. La dernière étape réalise la construction de la base dimensionnelle avec des sous étapes de tests et de validation par rapport aux besoins des utilisateurs. (De Miguel et al, 2000) a implanté un outil d'aide à la conception, appelé IDEA-DWCASE, qui supporte partiellement la méthode proposée.

L'avantage de cette méthode est la combinaison de l'approche descendante et ascendante ; elle intègre à la fois les données des sources opérationnelles et les besoins des décideurs dans la définition du modèle dimensionnel. Néanmoins, la première étape consacrée à la collecte des besoins des décideurs ne spécifie ni la manière dont ces besoins peuvent être exprimés, ni comment les transformer en schéma dimensionnel.

De son côté, (Trujillo et al, 2003) propose une méthode de conception de schémas dimensionnels basée sur le paradigme objet. Cette méthode utilise le standard UML pour la définition des concepts dimensionnels (cf. section 2.1.1.2 § GOLD). La méthode est basée sur quatre phases : analyse, conception, implantation et test. La phase d'analyse détermine

les besoins initiaux des décideurs en les interviewant, définit les règles de gestion et identifie les sources opérationnelles à partir desquelles les informations seront extraites. La phase de conception modélise le schéma conceptuel dimensionnel (GOLD) basé sur le paradigme objet, définit les processus d'extraction, de transformation et de chargement des données à partir des sources et finalement, conçoit les rapports d'analyse pour les décideurs. La phase d'implantation réalise la transformation du modèle conceptuel en modèle ROLAP, OOLAP ou MOLAP, définit les processus d'exportation des données et implante les rapports d'analyse. La dernière phase permet de valider la base décisionnelle en la confrontant aux besoins des décideurs.

4.4. Bilan

Dans le tableau suivant, nous comparons ces méthodes en nous basant sur les critères suivants :

- Les niveaux d'abstraction. Ces méthodes sont caractérisées par leur décomposition en trois niveaux d'abstractions. Ces niveaux d'abstraction, recommandés par ANSI/X3/SPARC, distinguent les niveaux conceptuel, logique et physique. Le niveau conceptuel permet de garder un haut niveau d'abstraction en décrivant les données indépendamment de toutes spécificités techniques. Le niveau logique permet de relier cette abstraction à un modèle dimensionnel connu (ROLAP, MOLAP, HOLAP ou OOLAP). Finalement, le niveau physique permet de tenir compte de l'outil utilisé pour l'implantation du schéma et de définir l'ensemble des optimisations possibles sur une plate forme particulière ;
- La catégorie de la méthode en fonction de la démarche appliquée : ascendante basée sur les données sources, descendante basée sur les besoins décideurs ou mixte combinant les deux démarches ;
- La formalisation de la démarche : définition d'un ensemble d'étapes bien définies, ordonnées et formalisées ;
- La définition d'un modèle conceptuel pour la modélisation des besoins décisionnels, un modèle comporte un ensemble de concepts et de formalismes ;
- La proposition d'un outil d'aide à la conception, un outil basé sur la méthode proposée.

Critères / Travaux	Niveaux d'abstraction			Catégorie			Démarche	Modèle		Outil
	Conceptuel	Logique	Physique	Descendante	Ascendante	Mixte		Concept	Formalisme	
(Kimball et al, 2002)		√	√	√			√	√		
(Tsois et al, 2001)	√	√		√				√	√	
(Prat et al, 2002)	√	√	√	√				√		
(Golfarelli et al, 1998)	√	√	√		√		√	√	√	√
(Cabibbo et al, 1998)	√	√			√			√	√	
(Moody et al, 2000)	√	√			√		√	√		
(Carneiro et al 2002)	√	√	√			√	√			√
(Cavero et al, 2001)	√	√	√			√	√	√		√
(Trujillo et al, 2003)	√	√	√			√	√	√	√	√

***Tableau I.6** : TABLEAU COMPARATIF DES TRAVAUX SUR LES METHODES*

Les méthodes proposées se reposent sur un modèle dimensionnel suivant un paradigme objet (Trujillo et al, 2003) (Moody et al, 2000), relationnel (Kimball et al, 2002) ou purement dimensionnel (Cabibbo et al, 1998). Une première catégorie de ces méthodes propose une démarche descendante qui se base uniquement sur les besoins des décideurs. Une deuxième catégorie propose une démarche ascendante basée uniquement sur les données des sources, négligeant le rôle du décideur dans le processus de construction de la base décisionnelle. Une troisième catégorie de méthodes mixtes intègre les données des sources aux spécifications des besoins des décideurs lors de la modélisation de la base décisionnelle permettant ainsi d'exploiter les données existantes et de fournir une réponse succincte et de qualité aux requêtes des décideurs.

5. Notre proposition

Dans cette section, nous présentons le cadre général de nos travaux de recherche, notamment l'architecture de notre système décisionnel. Puis, nous exposons la problématique qui a motivé ce travail de thèse et qui concerne la conception et la manipulation des données dimensionnelles. Enfin, nous présentons les objectifs de nos travaux de recherches.

5.1. Cadre général

Au sein de notre équipe, nous avons proposé une architecture de systèmes décisionnels basée sur la séparation de l'entrepôt et des magasins de données ayant des objectifs différents et des problèmes à résoudre divergents (Teste, 2000). L'**entrepôt** regroupe toute l'information décisionnelle tandis que les **magasins** contiennent une partie de cette information, dédiée à un thème, un métier ou une analyse (cf. Figure I.2). Contrairement à la plupart des propositions faites dans la littérature, nous n'adoptons pas une modélisation dimensionnelle au niveau de l'entrepôt en considérant que son objectif est de faciliter la gestion efficace des données et leur historisation. Par contre, les magasins se basent sur une structure dimensionnelle plus adaptée aux analyses afin de se rapprocher de la manière dont les décideurs perçoivent les données analysées (Codd et al, 1993).

Nous situons nos travaux de recherche dans le cadre de la modélisation des magasins de données dimensionnelles. Dans un premier temps, nous souhaitons nous focaliser sur la **modélisation** des données dimensionnelles et l'**interrogation** interactive, fiable et rapide de ces données. Dans un second temps, nous désirons spécifier une **démarche de conception** des schémas dimensionnels à partir de l'entrepôt de données historisées dans le cadre de notre architecture de systèmes décisionnels.

5.2. Existant et limites

Plusieurs travaux ont proposé des modèles de données dimensionnelles. Pourtant aucun de ces modèles n'est unanimement accepté et aucun standard n'a été proposé jusqu'à ce jour.

Les modèles proposés formalisent les concepts dimensionnels de différentes manières et présentent parfois des langages pour manipuler les données analysées. Ils utilisent des bases relationnelles, objets ou purement dimensionnelles pour sauvegarder les données. Néanmoins, peu d'auteurs s'intéressent au niveau conceptuel dans la

modélisation dimensionnelle (Golfarelli et al, 1998) (Tryfona et al, 1999) (Lujan et al, 2004) (Trujillo et al, 2003).

La plupart de ces modèles proposent de définir des schémas en étoile basés sur la dualité fait - dimension. Cette modélisation en étoile ne facilite pas la corrélation entre les différents sujets d'analyse tels que, par exemple, la comparaison entre les ventes et les achats pour une même ville dans une application d'analyse commerciale.

Tous les travaux n'intègrent pas la définition de multi-hiérarchies au sein d'une même dimension. Or, la multiplicité des hiérarchies est une caractéristique du monde réel ; souvent les objets sont classés selon différents critères indépendants (tranches d'âge et adresse pour les clients par exemple). Cette multiplicité nécessite une modélisation rigoureuse qui permet d'éviter les conflits éventuels entre ces hiérarchies durant l'analyse. (Hurtado et al, 2002) a traité ces conflits en intégrant un ensemble de contraintes. Définies au niveau de chaque dimension, ces contraintes ne gèrent pas les conflits qui peuvent exister entre les hiérarchies de différentes dimensions. L'intégration de ces contraintes a des répercussions sur la manipulation des données dimensionnelles lors des interrogations OLAP.

Lors de l'implantation des modèles dimensionnels, une deuxième catégorie de travaux s'est focalisée sur les techniques d'optimisation des interrogations des données dimensionnelles, telles que la matérialisation des vues et la création de nouveaux index.

La plupart de ces travaux n'intègrent pas dans leur solution la structure du modèle dimensionnel. Cependant, la prise en compte de la structure des hiérarchies dans les dimensions permet d'optimiser la sélection des vues (Baralis et al, 1997) (Paraboschi et al, 2003) et de faciliter les recherches qui en découlent (maintenance, calcul des requêtes). L'intégration des contraintes sémantiques exprimées au niveau du modèle conceptuel dans le processus de sélection des vues matérialisées permet de ne garder dans le processus de sélection que les vues cohérentes qui ne violent pas l'intégrité des contraintes sémantiques exprimées dans le modèle.

Dans le contexte décisionnel, la définition des concepts dimensionnels ne suffit pas pour permettre aux concepteurs de répondre aux besoins des décideurs. Une méthode de conception s'avère cruciale afin de définir les étapes à suivre et de fournir les outils nécessaires à la conception de la base dimensionnelle (Trujillo et al, 2003). Nous avons présenté dans la section 4.4 un comparatif de ces travaux qui décrivent des méthodes basées sur un modèle dimensionnel suivant une démarche ascendante, descendante ou mixte. Néanmoins, ces méthodes ne proposent pas d'outils pour la spécification et la formalisation des besoins décisionnels. En effet, nous retrouvons souvent dans ces méthodes une étape de '***collecte de besoins***' qui n'explicite pas au concepteur le processus à entreprendre pour spécifier ces besoins. Or, de ces besoins dépendent la bonne qualité et l'efficacité du système décisionnel.

5.3. Objectifs

Les travaux présentés dans cette thèse visent à étudier la modélisation des données décisionnelles afin de proposer un modèle dimensionnel au niveau conceptuel intégrant l'expression de contraintes structurelles et sémantiques, un langage de manipulation adapté et une méthode de conception des bases de données dimensionnelles contraintes.

Nous souhaitons proposer un modèle dimensionnel facilitant la corrélation de plusieurs sujets d'analyse. Aussi, ce modèle doit intégrer le concept de constellation avec une multi-hiérarchisation des dimensions.

Afin d'assurer l'intégrité des données et la cohérence des analyses décisionnelles, nous désirons intégrer dans notre modèle l'expression des contraintes structurelles et sémantiques. L'expression de ces contraintes dans le modèle dimensionnel permet de spécifier les interactions entre les instances des dimensions aux hiérarchies multiples afin d'éviter les conflits lors des interrogations des données dimensionnelles (notamment lors des opérations de forage et de rotation).

Au niveau de la manipulation des données, nous souhaitons étudier l'impact des contraintes sur les opérateurs dimensionnels afin de permettre au décideur de mieux spécifier ses besoins en précisant l'ensemble des instances à analyser. Ainsi, nous allons nous centrer sur la définition d'un langage de manipulation des données dimensionnelles intégrant les contraintes.

Nous proposons d'étudier également l'impact de ces contraintes sur la construction des bases de données dimensionnelles. En effet, l'intégration des contraintes dans le calcul des pré-agrégats (appelés vues matérialisées) permet d'éliminer toutes les vues incohérentes.

De plus, la définition d'un schéma dimensionnel qui répond aux besoins des décideurs et qui intègre l'expression des contraintes nécessite la mise au point d'une méthode de conception des bases de données dimensionnelles. Aussi, nous souhaitons proposer une méthode de conception intégrant à la fois la description des données sources et la définition des besoins décideurs. Afin de faciliter la tâche au concepteur et de réaliser une modélisation fiable et efficace, nous expliciterons dans notre méthode le processus de collecte des besoins des décideurs basé sur un ensemble de sous-étapes progressives et incrémentales. La méthode doit comporter des sous-étapes facilitant la détection et la formalisation des contraintes sémantiques dans le schéma dimensionnel.

Afin de valider nos travaux, nous souhaitons proposer un outil d'aide à la conception de bases de données dimensionnelles. Cet outil doit permettre d'assister le concepteur lors de la construction graphique des magasins de données dimensionnelles à partir de l'entrepôt de données.

CHAPITRE II : PROPOSITION D'UN MODELE DIMENSIONNEL CONTRAINT

PLAN DU CHAPITRE

Dans ce chapitre, nous présentons notre modèle conceptuel pour un magasin de données dimensionnelles à contraintes. Nous avons choisi d'adopter le modèle dimensionnel au niveau des magasins de données afin de faciliter l'interrogation et l'analyse des données décisionnelles.

Dans un premier temps, nous présentons notre problématique et les apports de notre modèle. La deuxième section présente les concepts fondamentaux de notre modèle dimensionnel intégrant l'expression des contraintes. La troisième section décrit la gestion du temps dans notre modèle. La quatrième section décrit les contraintes structurelles et sémantiques exprimées dans ce modèle.

1. Introduction à la modélisation dimensionnelle

La simplicité des structures de données présente l'un des principaux critères de réussite de l'approche OLAP, basée généralement sur un modèle dimensionnel. Ce modèle organise les données en sujets d'intérêt (faits) analysés en fonction de différents axes (dimensions) organisés en différents niveaux de granularités (hiérarchies). Cette organisation dimensionnelle permet une manipulation et une exploitation des données à des fins décisionnelles, d'une manière rapide, efficace et performante (Codd et al, 1993) (Kimball et al, 2002).

1.1. Problématique

En l'absence d'un modèle consensuel pour les données dimensionnelles, plusieurs propositions et formalisations ont été présentées par différents auteurs donnant lieu à une prolifération de notations et de définitions (Gyssen et al, 1997) (Lehner, 1998) (Cabibbo et al, 1998) (Bellahsène et al, 1999) (Pedersen et al, 1999) (Trujillo et al, 2003).

Notre objectif est de proposer un modèle dimensionnel qui palie les limites des modèles existants. Notamment, nous souhaitons que notre modèle :

- fasse abstraction des contraintes techniques de stockage de données afin de s'approcher de la vision des décideurs (Golfarelli et al, 2002) ;
- facilite les corrélations entre les différents sujets d'analyse et spécifie les fonctions d'agrégations compatibles pour chaque indicateur d'analyse (Sapia et al, 1999) (Abelló et al, 2002) (Trujillo et al, 2003) ;
- intègre la définition d'axes d'analyse à perspectives multiples d'une manière explicite. La spécification des différents niveaux hiérarchiques doit intégrer la définition d'attributs faibles afin de compléter la sémantique des paramètres d'analyse ;
- permette l'intégration d'informations cohérentes et fiables nécessaires à des prises de décisions judicieuses ayant un impact sur l'avenir de l'organisation ou de l'entreprise. En effet, la conception d'un modèle de données décisionnelles fiables et cohérentes implique l'intégration de contraintes dans le modèle dimensionnel (Samtani et al, 1998) (Hurtado et al, 2002). La plupart des modèles dimensionnels s'intéressent aux contraintes structurelles. Ces modèles traitent le problème d'additivité des mesures le long des hiérarchies (Lehner, 1998) (Hurtado et al, 2002) (Lechtenbörger et al, 2002). Seuls les modèles de (Carpani et al, 2001) et de (Hurtado et al, 2002) intègrent les contraintes sémantiques ;

- gère les données temporelles à deux niveaux : détaillé et résumé, afin de ne garder que l'information pertinente à l'analyse. Quelques modèles proposent de définir des concepts dimensionnels temporels en intégrant la gestion du temps dans les faits et les dimensions (Pedersen et al, 1999) (Bellahsène, 2002) (Mendelzon et al, 2003). Aucun de ces travaux n'offre un mécanisme d'archivage des données anciennes devenues obsolètes permettant de réduire le volume des données analysées (Ravat et al, 1999) (Teste, 2000).

1.2. Notre proposition

Nous proposons un modèle dimensionnel qui organise les données d'une manière plus adaptée aux décideurs non informaticiens. Ce modèle se situe au niveau conceptuel afin de se détacher de toutes les spécificités d'ordre logique (ROLAP, MOLAP, OOLAP) et physique. Le choix du niveau conceptuel permet d'exprimer le choix de gestion indépendamment des moyens à mettre en œuvre et de leur organisation (Golfarelli et al, 1998). Ce modèle représente les données en une **constellation** de **faits** associés à des **dimensions** pouvant être partagées. Ainsi, notre modèle supporte l'expression de faits multiples, facilitant la corrélation entre les sujets d'analyse, et la définition explicite des hiérarchies multiples au sein des dimensions. La sémantique des attributs est complétée par l'intégration des fonctions d'agrégations compatibles au niveau de chaque mesure et d'attributs faibles décrivant les paramètres.

Notre modèle assure la cohérence et la fiabilité des données dimensionnelles en intégrant un ensemble de contraintes. Nous distinguons les contraintes suivantes :

- ***les contraintes structurelles*** sont définies sur les concepts dimensionnels à deux niveaux (structure et instance). Le respect des contraintes structurelles permet d'agréger les données analysées selon les différentes granularités offertes (Lehner, 1998) (Hurtado et al, 2002) (Lechtenborger et al, 2002) ;
- ***les contraintes sémantiques*** sont définies entre les hiérarchies *intra* et *inter-dimensions*. Ce type de contraintes se base sur la sémantique de l'application analysée. Ces contraintes, exprimées par le concepteur du schéma dimensionnel, sont extraites des règles de gestion de l'environnement analysé ;

L'originalité de notre approche réside dans la proposition d'une typologie de contraintes permettant d'identifier clairement les différentes catégories d'incohérences pouvant survenir. Nos travaux se démarquent de ceux de (Carpani et al, 2001) et (Hurtado et al, 2002), puisque non seulement nous permettons l'expression de contraintes sur les hiérarchies d'une dimension, mais nous permettons également de définir des contraintes inter-dimensions.

La définition des contraintes nous a permis de représenter simplement la dimension *Temps* qui analyse à la fois les données détaillées et les données archivées des faits. L'archivage des données du fait permet de réduire ses données jugées trop volumineuses (Kimball et al, 2002) et de ne garder que l'information pertinente à l'analyse.

Dans la section suivante, nous présentons les concepts de notre modèle dimensionnel contraint.

2. Modèle dimensionnel contraint

Cette section définit notre modèle dimensionnel basé sur les concepts de dimension multi-instanciable, de hiérarchie, de fait et de constellation.

2.1. Dimension et hiérarchie

Un des objectifs des systèmes décisionnels est de calculer la performance d'une entreprise ou d'une organisation. Cette performance est formalisée par des indicateurs. Chaque groupe d'indicateurs est relatif à un sujet d'analyse. Les dimensions représentent les axes d'analyse en fonction desquels sont manipulés les sujets d'analyse. Une dimension est formée d'attributs décrivant les caractéristiques des indicateurs d'analyse. Les attributs d'une dimension peuvent être organisés en hiérarchies, de la granularité la plus fine à la plus générale.

Soient :

- *N* un ensemble de noms,
- *ID* un ensemble d'identifiants,
- $DOM = \cup Dom_i$ où chaque Dom_i est un domaine (tels que entier, réel, caractère, chaîne de caractères, …) ; tout élément de *DOM* est une valeur.
- $(E, \preceq)$ un ensemble muni d'une fonction d'ordre. On dit que (E, ≤) est un *treillis* (ou un *lattis*) si toute partie ayant au moins deux éléments de E admet une borne inférieure et une borne supérieure. Par exemple, l'ensemble des sous-groupes d'un groupe donné, ordonné par l'inclusion, est un treillis : la borne inférieure est donnée par l'intersection, la borne supérieure par l'union des sous-groupes engendrés.

Définition

*Une **dimension** D est définie par* (N^D, A^D, H^D, I^D) *où :*

- $N^D \in N$ *est le nom de la dimension,*
- $A^D = \{a_1, a_2, \ldots, a_u\}$ *est un ensemble d'attributs,*
- $H^D = \{h^D_1, h^D_2, \ldots, h^D_v\}$ *est un ensemble de hiérarchies,*
- $I^D = \{I^D_1, I^D_2, \ldots\}$ *est l'ensemble des instances de D. Une instance est définie par le n-uplet* $[a_1{:}v_1, \ldots, a_u{:}v_u]$ *tel que* $\forall k \in [1..u], a_k \in A^D \wedge (v_k \in DOM \vee v_k \in ID)$.

Parmi les attributs d'une dimension, on distingue les attributs *All* et *Id* tels que *Dom(All)={'all')* et $Dom(Id) \in ID$. L'attribut Id est l'identifiant de la dimension. *All* désigne la granularité de plus haut niveau tandis que Id représente la granularité la plus fine.

Plusieurs ***hiérarchies*** d'attributs peuvent être définies au sein d'une même dimension. Ces hiérarchies représentent les perspectives d'analyse. Elles permettent de déterminer les niveaux de granularité auxquels peuvent être manipulées les indicateurs d'analyse.

Définition

*Une **hiérarchie** h^{Di}, définie sur une dimension D_i, est un chemin élémentaire acyclique débutant par Id et se terminant par All. Elle est définie par (N^h, $Param^h$, $Suppl^h$, $Cond^h$) où :*

- *$N^h \in N$ est le nom de la hiérarchie,*
- *$Param^h : P^D /\{All\} \rightarrow P^D /\{All\}$ ($P^D \subseteq A^D$) est une application décrivant la hiérarchie des attributs (chaque attribut est appelé paramètre de la hiérarchie) avec $P_i \rightarrow P_j$ implique que P_i est de granularité strictement plus fine que P_j,*
- *$Suppl^h : P^D \rightarrow A^D\text{-}P^D$ est une fonction décrivant les attributs faibles (attributs de la dimension n'appartenant pas à $Param^h$) associés à chaque paramètre,*
- *$Cond^h$ est une expression booléenne définissant la condition d'appartenance des instances de la dimension à une hiérarchie.*

Nous notons :

- $P^D \subseteq A^D$ l'ensemble des paramètres de la dimension *D*.
- $Param^D : P^D \rightarrow P^D$ est une fonction qui généralise $Param^h$ définissant l'ordre des paramètres P^D dans la dimension *D*. $Param^D (p_i)$ renvoie l'ensemble des paramètres de granularité moins fine relié à *pi;*
- $IParam^h : DOM(P^D) \rightarrow DOM(P^D)$ l'extension de l'application $Param^h$ définie sur les instances des paramètres de la hiérarchie avec $DOM(P^D)$ l'union des domaines des paramètres de la dimension. Cette application définit un ordre partiel sur les instances des paramètres de la hiérarchie ;
- $IParam^D : DOM(P^D) \rightarrow DOM(P^D)$ l'extension de la fonction $IParam^h$ définissant un ordre partiel sur les instances des paramètres de la dimension ;
- $I^D{}_k \in_{(cond)} h^D{}_i$ pour indiquer que l'instance k de I^D satisfait la condition $Cond^h$ et par conséquent $I^D{}_k$ appartient à la hiérarchie $h^D{}_i$.

Formalisme graphique. Nous proposons un formalisme graphique des différents concepts en adaptant le formalisme défini par (Golfarelli et al, 1998) aux spécificités de notre modèle.

Une dimension est représentée par un rectangle comportant le nom de la dimension et relié à un treillis ***(P^D, $Param^D$)*** représentant les hiérarchies de la dimension. Ce treillis est défini sur l'ensemble des paramètres de la dimension, P^D, associé à la fonction d'ordre des paramètres, $Param^D$. Ce treillis comporte comme racine (borne inférieure) le paramètre identifiant de la dimension (*Id)* et comme nœud final (borne supérieure) le paramètre *All*. Chaque hiérarchie est représentée par un chemin dans ce treillis. Chaque nœud dans ce chemin, schématisé par un cercle étiqueté, représente un paramètre de la hiérarchie. Les attributs faibles sont représentés par leurs noms et sont reliés aux paramètres qu'ils décrivent. Les différentes hiérarchies sont nommées. Le nom de chaque hiérarchie contenu dans un rectangle est placé après le dernier nœud commun de façon à mettre en relief les différents chemins du treillis représentant les hiérarchies.

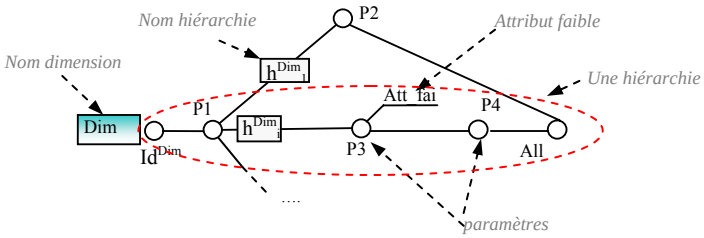


***Figure II.1** : Formalisme graphique d'une dimension et de ses hiérarchies*

Par abus de notation et afin de simplifier les exemples d'illustration, nous désignons les objets du modèle (dimensions, hiérarchies, …) par leur nom.

Exemple 1

Une société de location de voitures désire étudier la productivité de ses différentes agences de location au travers d'une application OLAP. Elle a besoin d'effectuer l'analyse quotidienne des locations de véhicules en fonction d'une offre proposée aux clients dans les différentes agences. L'analyse des locations de véhicules doit s'effectuer selon les axes d'analyse : *Agences*, *Clients*, *Véhicules* et *Temps*.

L'agence est caractérisée par son code, sa raison sociale et sa localisation. A ce niveau, nous souhaitons définir trois perspectives d'analyse en fonction de la localisation de l'agence. La première perspective décrit les agences suivant l'organisation géographique de la France en ville, département et région. La deuxième perspective, relative à l'organisation géographique des Etats-Unis, organise les villes par état. Enfin, la troisième perspective, commune à la France et aux Etats-Unis, décrit la position géographique des villes dans leur pays selon l'indication nord, sud, est, ouest.

Pour exprimer ces besoins, nous définissons la dimension *Agences* comme suit :

- $N^{Agences}$ = "Agences",
- $P^{Agences}$ = {CodeAg, Raison, Ville, Département, Nom_dpt, Région, Etat, Zone, Pays, All},
- $H^{Agences}$ = {geo_fr, geo_us, geo_zn},
- $I^{Agences}$ = {$I^{Agences}_1$, $I^{Agences}_2$, $I^{Agences}_3$,… }.

Nous présentons trois exemples d'instances de notre dimension *Agences* : deux agences françaises et une agence américaine. Chaque instance est un n-uplet de la forme suivante :

- $I^{Agences}_1$ = [*CodeAg* : 1, *Raison* : "Agence Campus31", *Ville* : "Toulouse", *Departement* : 31, *Nom_dpt* : "Hte-Garonne", *Région* : "Midi-Pyrénées", *Etat* : NULL, *Zone* : 'Sud-Fr', *Pays* : "France", *All* : "all"],
- $I^{Agences}_2$ = [*CodeAg* : 2, *Raison* : "Agence du Bouchon", *Ville* : "Lyon", *Département* : 69, *Nom_dpt* : "Rhône", *Région* : "Rhône-Alpes", *Etat* : NULL, *Zone* : 'Est-Fr', *Pays* : "France", *All* : "all"],
- $I^{Agences}_3$ = [*CodeAg* : 3, *Raison* : "Big Appel Agency", *Ville* : "New York", *Département* : NULL, *Nom_dpt* : NULL, *Région* : NULL, *Etat* : "New York", *Zone* : 'Ouest-EU', *Pays* : "Etats-Unis", *All* : "all"].

Pour compléter la définition de la dimension *Agences*, nous définissons trois hiérarchies. La hiérarchie "*geo_fr*" décrit les agences suivant l'organisation

géographique française tandis que la hiérarchie "*geo_us*" est relative à l'organisation géographique des Etats-Unis. Enfin, la hiérarchie "*geo_zn*" décrit la perspective d'analyse par zone.

La hiérarchie "*geo_fr*", par exemple, comporte les paramètres *CodeAg*, *Ville*, *Département*, *Région*, *Pays* et *All*. Ces paramètres sont reliés par une fonction d'ordre les organisant du niveau le plus fin (*CodeAg*) au niveau le moins fin d'analyse (*All*). Les attributs faibles *Raison* relié à *CodeAg* et *Nom_dpt* relié au *Département,* complètent la sémantique de ces paramètres. La condition d'appartenance à cette hiérarchie est définie par le prédicat *Pays* = "*France*".

- $h^{Agences}_1$ = ("geo_fr", {$Param^{geo\_fr}$(CodeAg) = Ville, $Param^{geo\_fr}$(Ville) = Département, $Param^{geo\_fr}$(Département) =Région, $Param^{geo\_fr}$(Région) =Pays, $Param^{geo\_fr}$(Pays) = All}, {$Suppl^{geo\_fr}$(CodeAg) = {Raison}, $Suppl^{geo\_fr}$(Département) = {Nom_dpt}}, Pays = "France"),
- $h^{Agences}_2$ = ("geo_us", {$Param^{geo\_us}$(CodeAg) = Ville, $Param^{geo\_us}$(Ville) = Etat, $Param^{geo\_us}$(Etat) = Pays, $Param^{geo\_us}$(Pays) = All}, {$Suppl^{geo\_us}$(CodeAg) = {Raison}}, Pays = "Etats-Unis" $\wedge$ Etat $\neq$ NULL),
- $h^{Agences}_3$ = ("geo_zn", {$Param^{geo\_zn}$(CodeAg) = Ville, $Param^{geo\_zn}$(Ville) = Zone, $Param^{geo\_zn}$(Zone) = Pays, $Param^{geo\_zn}$(Pays) = All}, {$Suppl^{geo\_zn}$(CodeAg) = {Raison}}, Zone $\neq$ NULL).

La spécificité de multi-instanciation de notre modèle réside dans l'intégration d'une condition d'appartenance des instances de la dimension aux hiérarchies (la propriété $Cond^h$). Ainsi, les instances {$I^{Agences}_1$, $I^{Agences}_2$} appartiennent à "*geo_fr*" tandis que l'instance {$I^{Agences}_3$} appartient à "*geo_us*" et les instances {$I^{Agences}_1$, $I^{Agences}_2$, $I^{Agences}_3$} appartiennent à "*geo_zn*". Dans notre modèle, ceci est exprimé par les expressions suivantes :

- $I^{Agneces}_1, I^{Agences}_2 \in_{(cond)}$ *geo_fr*
- $I^{Agences}_3 \in_{(cond)}$ *geo_us*
- $I^{Agences}_1, I^{Agences}_2, I^{Agences}_3 \in_{(cond)}$ *geo_zn*.

La figure suivante présente la représentation graphique de la dimension *Agences*.

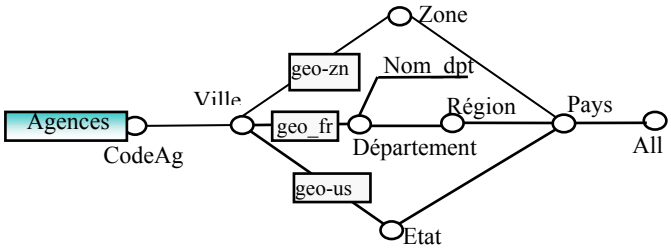


***Figure II.2** : Représentation graphique de la dimension Agences et de ses hiérarchies*

2.2. Fait

Tout sujet d'analyse est représenté par un fait formant un centre d'intérêt pour les décideurs. Chaque fait est caractérisé par une ou plusieurs mesures représentant les indicateurs analysés. Les mesures sont généralement de type numérique afin de pouvoir les agréger en fonction des axes de l'analyse.

Définition

*Un **fait** F est défini par (N^F, M^F, I^F, $IStar^F$) où :*

- *$N^F \in N$ est le nom du fait,*
- *$M^F = \{(m_1, \mho_1), (m_2, \mho_2),\ldots, (m_w, \mho_w)\}$ est un ensemble de couples mesures (ou indicateurs) et fonctions d'agrégations compatibles avec chaque mesure avec $\mho_i \subseteq$ {sum, max, min, avg, count},*
- *$I^F = \{I^F_1, I^F_2,\ldots\}$ est l'ensemble des instances de F. Une instance est définie par le n-uplet $[m_1:v_1, m_2:v_2,\ldots, m_w:v_w]$ où $\forall k \in [1..w], m_k \in M^F \wedge v_k \in DOM$.*
- *$IStar^F$ est une fonction associant chaque instance de I^F à une instance de chaque dimension liée au fait.*

Formalisme graphique. Un fait est représenté par un rectangle divisé en deux. La partie supérieure contient le nom du fait et la partie inférieure les noms des mesures.

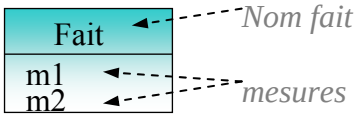


***Figure II.3** : Formalisme graphique d'un fait*

Exemple 2

Nous reprenons l'exemple de l'analyse des locations. Nous souhaitons que le sujet d'analyse Location contiennent deux indicateurs d'analyse : le montant de chaque location de véhicule et le nombre de jours de location. Ainsi, le fait *Location* est constitué de deux mesures *montant* et *nbjours*. Ce fait peut être spécifié de la manière suivante :

- $N^{Location}$ = "Location",
- $M^{Location}$ = {(montant, {sum, avg}), (nbjours, {sum, avg})},
- $I^{Location} = \{I^{Location}_1, I^{Location}_2,\ldots\}$,
- la fonction $IStar^{Location}$ est définie par $\{I^{Location}_1 \rightarrow \{I^{Temps}_1, I^{Clients}_1, I^{Agences}_1, I^{Vehicules}_1\}$, $I^{Location}_2 \rightarrow \{I^{Temps}_1, I^{Clients}_1, I^{Agences}_1, I^{Vehicules}_2, \ldots\}, \ldots.\}$.

La fonction $IStar^{Location}$ permet de relier une instance du fait aux instances des dimensions qui le déterminent. Ainsi, l'instance du fait *Location* $I^{Location}_1$ est déterminée par les instances I^{Temps}_1, $I^{Clients}_1$, $I^{Agences}_1$ et $I^{Vehicules}_1$ des dimensions respectives *Temps*, *Clients*, *Agences* et *Véhicules*. Une instance du fait est un n-uplet, qui comporte les valeurs des mesures *montant* et *nbjours*, représentée de la forme suivante :

- $I^{Location}_1$ = [montant : 540.00, nbjours : 8],
- $I^{Location}_2$ = [montant : 1200.00, nbjours : 22].

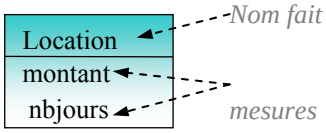


***Figure II.4** : Représentation graphique du fait Location*

2.3. Constellation

Le modèle en constellation est une généralisation du modèle en étoile (Kimball et al, 2002). Une constellation regroupe plusieurs sujets d'analyse (faits) étudiés selon différents axes d'analyse (dimensions) éventuellement partagés. La représentation de plusieurs sujets d'analyse dans le même schéma dimensionnel, facilite la corrélation des analyses telle que la comparaison des chiffres d'affaires réalisés par les employés dans les agences de location et le montant des locations réalisé dans celles-ci.

Nous intégrons dans la définition de la constellation une propriété qui décrit l'ensemble des contraintes sémantiques relatives aux données de cette constellation. Une plus ample description de ces contraintes est présentée dans la section 4.

Définition

*Une **constellation** C est définie par (N^C, F^C, D^C, $Star^C$, $Cons^C$) où*

- *$N^C \in N$ est le nom de la constellation,*
- *$F^C = \{F_1, F_2,\ldots, F_p\}$ est un ensemble de faits,*
- *$D^C = \{D_1, D_2,\ldots, D_q\}$ est un ensemble de dimensions,*
- *$Star^C : F^C \rightarrow D^C$ est une fonction associant les faits aux dimensions afin de spécifier les sujets d'analyse et les axes d'études associés,*
- *$Cons^C$ représente les contraintes sémantiques associées à la constellation (cf. section 4.2).*

Formalisme graphique. Une constellation est composée d'un ensemble de faits et de dimensions. Chaque dimension est reliée à un ou plusieurs faits.

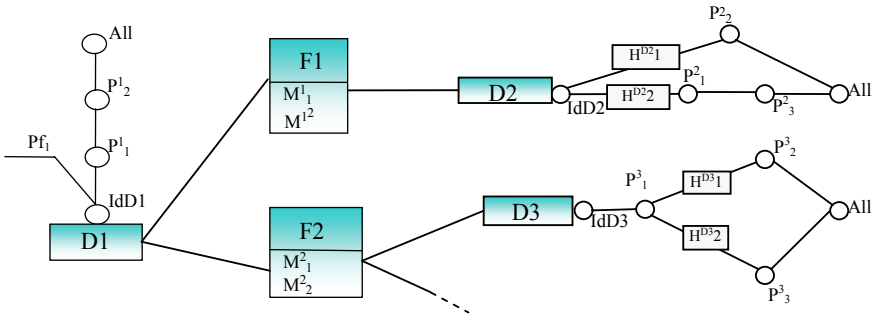


***Figure II.5** : Formalisme graphique d'une constellation*

Exemple 3

La société de location de voitures souhaite, en plus, corréler les analyses des performances des agences (nombre de locations, montant des locations) avec les performances des employés de chaque agence (chiffres d'affaires, marge). L'analyse des performances des employés est réalisée en fonction des axes d'analyse : *Temps, Employés et Agences*.

Ce besoin peut être traduit par un modèle dimensionnel organisé selon une constellation comportant deux faits (*Location, PERF*) et cinq dimensions (*Temps, Clients, Véhicules, Agences, Employés*). La constellation peut être définie par (N^C, F^C, D^C, $Star^C$, $Cons^C$) où

- N^C = "Location Véhicule",

- F^C = {Location, PERF},
- D^C = {Temps, Clients, Agences, Employés, Véhicules},
- $Star^C$ = {Location →{Temps, Clients, Véhicules, Agences}, PERF→{Temps, Agences, Employés} },
- $Cons^C$ = {C_1, …}. Une illustration de ces contraintes est présentée tout au long de la section 4.2.

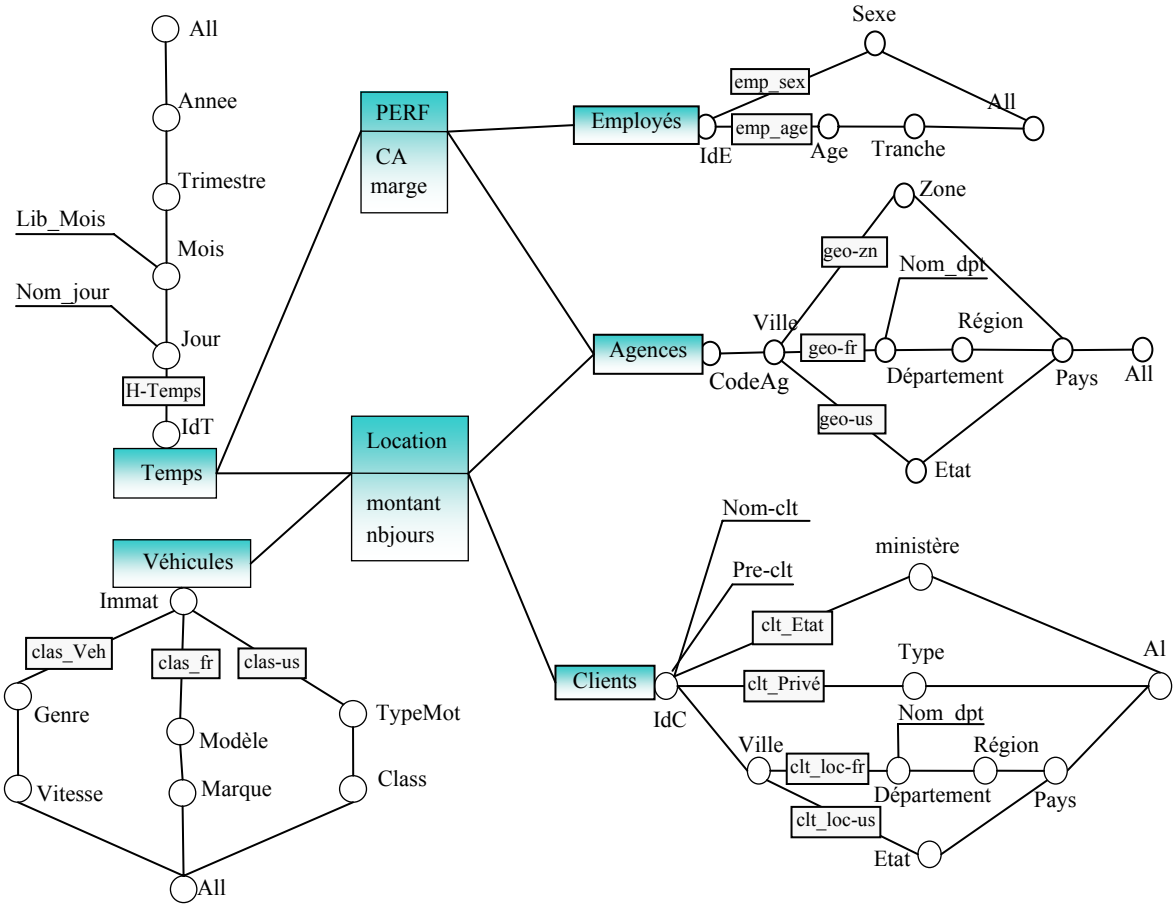


***Figure II.6** : Représentation graphique d'une constellation.*

3. Dimension temps

La dimension temps occupe une position importante dans les analyses décisionnelles (Yang et al, 2000) (Mendelzon et al, 2003). L'analyse de l'historique permet de prévoir les actions à entreprendre et d'anticiper sur les futurs événements (Mkaouar et al, 2003).

La plupart des modèles gèrent cette dimension de manière identique aux autres dimensions (Agrawal et al, 1997) (Cabibbo et al, 1998) (Kimball et al, 2002). D'autres y consacrent une gestion spécifique (Mendelzon et al, 2000) (Mendelzon et al, 2003) permettant de gérer l'évolution des données des dimensions. Cependant, tous ces modèles proposent de conserver le détail de l'historique sans considérer le volume des données qui se multiplient rapidement et la pertinence de l'information conservée durant de longues périodes.

Dans notre modèle, nous proposons de conserver les données dimensionnelles sous forme détaillée et/ou archivée. Étant donné le volume important que peut prendre

l'historique détaillé (Kimball et al, 2002), nous avons défini un processus d'archivage (Ravat et al, 2000a). Ce processus permet de résumer le détail des informations considérées comme très anciennes pour ne garder que l'information pertinente aux décideurs. L'intégration de cette gestion adaptée aux besoins décisionnels au niveau du modèle dimensionnel nécessite la définition d'une dimension temps qui décrit les deux niveaux de l'historique : détaillé et résumé.

Afin de répondre à ce besoin, nous proposons de définir deux hiérarchies dans la dimension *Temps* :

- une hiérarchie temporelle détaillée décrivant l'historique détaillé des données,
- une hiérarchie temporelle archivée décrivant les données archivées ou résumées.

Notre modèle permet de définir ces hiérarchies à l'aide de notre concept de dimension à instanciation multiple. En effet, les deux hiérarchies temporelles se différencient par l'ensemble des instances qu'elles gèrent et qui seront analysées suivant des granularités différentes. Notre modèle permet de les modéliser en utilisant la condition d'appartenance à une hiérarchie.

Exemple 4

Dans l'exemple de la 0, nous avons défini deux hiérarchies temporelles détaillées et archivées. Les faits Location et PERF sont reliés à cette dimension. La hiérarchie temporelle d'archives T_arch décrit les instances archivées du fait Location ; nous ajoutons le symbole Ⓗ à la représentation graphique de ce fait pour indiquer qu'il est archivé. La deuxième hiérarchie détaillée permet d'analyser les données de location au niveau jour (le plus détaillé) de l'année 1995 à ce jour. Les locations antérieures à 1995 sont archivées tous les trimestres. Les instances du fait PERF, n'étant pas archivées, sont analysées seulement selon la hiérarchie temporelle détaillée (cf. 0).

La dimension *Temps* peut être définie comme suit :

- N^{Temps} = "Temps",
- P^{Temps} = {IdT, Jour, Nom_jour, Mois, Lib_Mois, Trimestre, Année, All},
- H^{Temps} = {T_det, T_arch},
- I^{Temps} = {I^{Temps}_1, I^{Temps}_2, I^{Temps}_3, …}.

Avec

- h^{T_det} = ("T_det", {IdT→Jour, Jour→Mois, Mois→ Trimestre, Trimestre → Année, Année→All}, {Jour→{Nom_Jour}, Mois→{Lib_Mois}}, Jour <> Null),
- h^{T_arch} = ("T_arch", {IdT→Trimestre, Trimestre→Année, Année→All}, {}, Trimestre <> Null et Année < =1995)

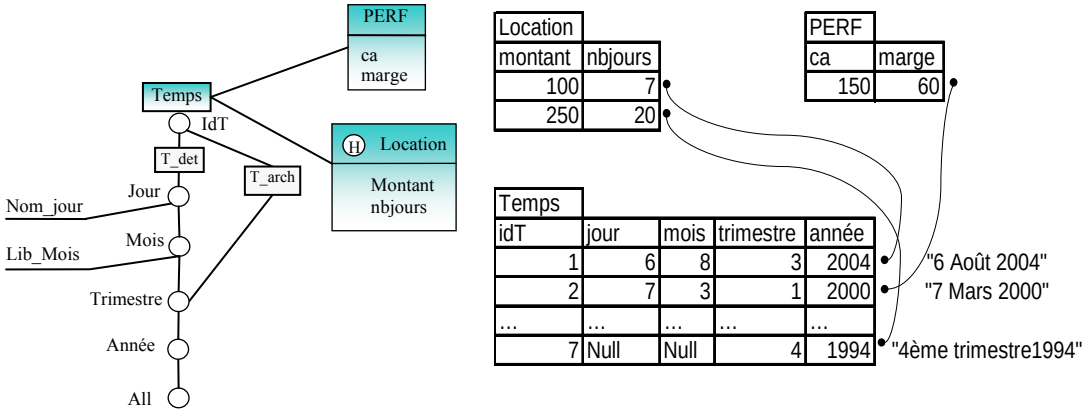


***Figure II.7** : Représentation des hiérarchies temporelles accompagnée d'un exemple d'instances.*

Entre les hiérarchies détaillées et archivées de la dimension temporelle, nous définissons des contraintes qui déterminent l'ensemble des faits correspondant à chaque période (détaillée et archivée). En effet, la spécificité de ces deux hiérarchies est qu'elles divisent l'axe du temps en deux parties : une première partie qui permet de dater des faits détaillés (niveau jour dans notre exemple) et une deuxième où les faits sont résumés et ne peuvent être analysés qu'à un niveau moins fin (tous les trimestres par exemple). Il faut noter que nous ne pouvons utiliser qu'une seule hiérarchie (détaillée) de la dimension *Temps* si le fait n'est pas archivé. C'est la condition d'appartenance définie dans notre modèle qui permet de spécifier la hiérarchie à appliquer avec chaque instance.

Le support de hiérarchies multiples à multi-instanciations nécessite l'intégration d'un ensemble de contraintes structurelles et sémantiques. Dans la section suivante, nous présentons ces contraintes qui permettent d'éviter les incohérences dans les analyses décisionnelles et d'exprimer les interactions entre les différentes hiérarchies d'une même ou de plusieurs dimensions.

4. Contraintes

La définition des contraintes dans les modèles de données permet de spécifier les règles de gestion de l'environnement de l'analyse. Cette spécification est nécessaire afin de garantir l'intégrité et la confidentialité des données. Ce besoin de fiabilité et de confidentialité est plus pressant dans les systèmes décisionnels. En effet, garantir la cohérence des données est nécessaire à la prise de décision ayant un impact sur la survie de l'organisation (Kimball et al, 2002). Dans le cadre de ces systèmes, les modèles dimensionnels basés sur des hiérarchies multiples nécessitent la spécification de contraintes sur la structure de ces hiérarchies (Hurtado et al, 2002) (Lechtenborger et al, 2002). Peu de modèles proposés vérifient la validité de ces contraintes touchant à la structure dimensionnelle (Mendelzon et al, 2000). En outre, la modélisation des règles de gestion relatives à la sémantique de l'application analysée nécessite l'intégration de contraintes sémantiques dans le modèle dimensionnel.

Pour répondre à ces besoins, nous définissons les contraintes suivantes :

Les contraintes structurelles sont définies sur les concepts et les instances du modèle de données. Elles interviennent lors de la validation de la structure des hiérarchies du schéma dimensionnel. Par exemple, il ne doit pas exister de cycle entre les paramètres

d'une même dimension. Ces contraintes permettent d'assurer une structure valide des hiérarchies afin d'agréger correctement les données lors des opérations de forage. Nous distinguons les contraintes structurelles suivantes :

- *Contraintes liées aux concepts*. Elles comportent les contraintes syntaxiques, englobant les contraintes d'intégrité classiques des bases de données (unicité de nom, concept non vide, ...) et les contraintes hiérarchiques permettant de spécifier des hiérarchies valides au sein d'une dimension (acyclicité, connexion, ...).
- *Contraintes liées aux instances*. Elles englobent les contraintes reliant les instances des données dimensionnelles. Notamment, les liens entre les instances des faits des dimensions et des hiérarchies.

Les contraintes sémantiques, liées au contexte d'analyse, sont définies lors de la construction d'un schéma dimensionnel par le concepteur lui-même et seront consultées par le système lors des opérations d'extraction et d'interrogation des données dimensionnelles. Nous définissons deux familles de contraintes sémantiques selon la portée de ces contraintes :

- *contraintes intra-dimensions* qui s'appliquent aux hiérarchies de la même dimension et agissent sur les instances de celle-ci ;
- *contraintes inter-dimensions* qui s'appliquent aux hiérarchies de différentes dimensions et agissent sur les instances du fait associées à ces dimensions.

Les sections suivantes présentent ces différents types de contraintes intégrées dans notre modèle dimensionnel. Dans la première section, nous définissons les contraintes structurelles comportant les contraintes liées aux concepts et celles liées aux instances. Dans la deuxième section, nous décrivons les contraintes sémantiques appliquées sur les hiérarchies de notre schéma dimensionnel aux niveaux intra- et inter-dimensions.

4.1. Contraintes structurelles

Les contraintes structurelles sont liées aux structures des concepts du modèle dimensionnel et à leurs instances. Elles sont complémentaires aux différentes définitions des concepts de notre modèle dimensionnel présentés précédemment (cf. section 2). Nous distinguons les contraintes structurelles suivantes :

Contraintes liées aux concepts. Ces contraintes sont appliquées au niveau des concepts dimensionnels. Elles permettent de vérifier si la structure des faits et des dimensions est valide. La structure des différentes hiérarchies doit respecter un ensemble de règles afin de permettre l'analyse dimensionnelle des données à différents niveaux de granularité.

Contraintes liées aux instances. Ces contraintes sont appliquées sur les instances des dimensions et des faits. La validation de ces contraintes nécessite la vérification des valeurs des données.

4.1.1. Contraintes liées aux concepts

Nous distinguons deux familles de contraintes liées aux concepts : les contraintes syntaxiques et les contraintes hiérarchiques.

Pour formaliser les contraintes, nous utilisons les notations mathématiques suivantes :

- les opérateurs existentiel, universel, de négation et d'unicité respectivement notés $\forall$, $\exists$, $\neg$, !,
- les connecteurs logiques d'union, d'intersection, d'implication et de double implication respectivement notés $\vee$, $\wedge$, $\Rightarrow$, $\Leftrightarrow$.
- l'opérateur de comptage $|f|$ calculant le nombre d'éléments dans un ensemble f.

4.1.1.1. Contraintes syntaxiques

Ces contraintes englobent toutes les contraintes d'intégrité classiques que l'on retrouve dans les bases de données. Nous citons :

♦ ***Unicité de nom***

Les noms des concepts de constellation N^C, de fait N^F, de dimension N^D et de hiérarchie N^h, définis dans le modèle, sont uniques. Les noms des paramètres N^{PDi} de chaque dimension D_i et des mesures N^{MFi} de chaque fait F_i sont uniques respectivement dans leur dimension et fait. Cette contrainte est définie par l'expression suivante :

$$\forall N^i \in N = \{N^C \cup N^F \cup N^D \cup N^h\},\ \forall N^j \in N \wedge i \neq j \Rightarrow N^i \neq N^j$$
$$\forall N^i \in N^{PDk},\ \forall N^j \in N^{PDk} \wedge i \neq j \Rightarrow N^i \neq N^j$$
$$\forall N^i \in N^{MFk},\ \forall N^j \in N^{MFk} \wedge i \neq j \Rightarrow N^i \neq N^j$$

♦ ***Dimension non vide***

Chaque dimension possède au moins deux paramètres : son identifiant (Id) et le paramètre All. Cette contrainte est définie par l'expression suivante :

$$\forall D_i \in D^C, Id \in P^{Di} \wedge All \in P^{Di}$$

♦ ***Fait non vide***

Chaque fait possède au moins une mesure. Cette contrainte est définie par l'expression suivante :

$$\forall F_i \in F^C, |M^{Fi}| >= 1$$

♦ ***Hiérarchie non vide***

Chaque hiérarchie possède au moins deux niveaux de paramètres. Dans notre définition de hiérarchie (cf § 2.1), ceci implique l'existence d'au moins un couple de paramètres vérifiant la fonction $Param^h$. Cette contrainte est définie par l'expression suivante :

$$\forall h^D \in H^D, D \in D^C, \exists p_n \in P^D, \exists p_m \in P^D \wedge p_n \neq p_m \Rightarrow Param^{h_D}(p_n) = p_m$$

♦ ***Fait non isolé***

Chaque fait est connecté à au moins une dimension. Cette contrainte est définie par l'expression suivante :

$$\forall F_i \in F^C, \exists D_j \in D^C \wedge D_j \in Star(F_i)$$

♦ ***Dimension non isolée***

Chaque dimension est connectée à au moins un fait. Cette contrainte est définie par l'expression suivante :

$$\forall D^j \in D^C, \exists\ F^i \in F^C \wedge D^j \in Star(F^i)$$

♦ ***Constellation non vide***

Chaque constellation comporte au moins un fait et une dimension. Cette contrainte est définie par l'expression suivante :

$$\forall C, D^C \neq \varnothing \wedge F^C \neq \varnothing$$

4.1.1.2. Contraintes hiérarchiques

Ces contraintes englobent les contraintes spécifiques à la structure hiérarchique des dimensions dans les modèles dimensionnels. Elles assurent la cohérence des analyses dimensionnelles notamment lors des opérations de forage entre les différents niveaux hiérarchiques.

♦ ***Connexion vers le haut***

Tous les paramètres, sauf All, possèdent au moins un père (un paramètre de granularité moins fine). Cette contrainte permet d'assurer le passage d'un niveau de détail à un autre. Dans notre modèle, les paramètres des hiérarchies vérifient cette contrainte qui est définie par l'expression suivante :

$$\forall p_i \in P^D, \exists p_j \in P^D \wedge p_i \neq All \Rightarrow p_j \in Param^D(p_i)$$

♦ ***Acyclicité***

Cette contrainte est définie entre les paramètres d'une même hiérarchie. Un paramètre ne peut pas être père et fils d'un autre paramètre. L'existence d'un cycle dans les hiérarchies est interdite.

Cette contrainte est définie par l'expression suivante :

$$\forall p_i, p_j \in P^D \wedge p_j \in Param^D(p_i) \Rightarrow p_i \notin Param^D(p_j)$$

♦ ***Connexion***

L'existence d'une relation de dépendance entre deux valeurs de paramètres (ex. Paris, France) implique l'existence d'un lien hiérarchique entre leur paramètre respectif (Ville, Pays). Cette contrainte est définie par l'expression suivante :

$$\forall v_i \in DOM(p_i),\ \forall v_j \in DOM(p_j) \wedge p_i \in P^D \wedge p_j \in P^D$$

$$v_j \in IParam^D(v_i) \Rightarrow p_j \in Param^D(p_i)$$

4.1.2. Contraintes liées aux instances

Ces contraintes font appel aux instances des dimensions et des faits.

4.1.2.1. Contrainte de dépendance fait-dimension

Considérons la fonction $Star^C$: $F^C \rightarrow D^C$ définie au niveau d'une constellation dans notre modèle dimensionnel. Cette fonction permet de relier chaque fait *F* à l'ensemble de ses dimensions D_i en respectant la contrainte de dépendance fonctionnelle entre les dimensions et le fait analysé.

Au niveau des instances, c'est la fonction $IStar^F$ associée à chaque instance du fait I^F_i qui vérifie la relation de dépendance fonctionnelle entre I^F_i et les instances des dimensions associées à I^F_i. Cette contrainte est définie par l'expression suivante :

$$\forall I^F_1, I^F_2 \in I^F \wedge IStar^F(I^F_1) = Istar^F(I^F_2) \Rightarrow I^F_1 = I^F_2$$

4.1.2.2. Contrainte de dépendance hiérarchique

La fonction $Param^D : P^D \rightarrow P^D$ définit un lien de dépendance hiérarchique entre les valeurs (instances) des paramètres de la dimension. Cette contrainte est définie par l'expression suivante :

$$\forall (v_1, v_2), (v_1', v_2') \in Dom(p_i) \times Dom(p_j) \wedge p_j \in Param^D(p_i),$$
$$v_1 = v_1' \Rightarrow v_2 = v_2'$$

4.1.2.3. Contrainte de partition

A chaque valeur d'un paramètre correspond une et une seule valeur parmi les valeurs de chaque paramètre successeur dans la hiérarchie. Par exemple, la ville de Toulouse qui appartient à la France ne peut pas être reliée à un autre pays. Cette contrainte est définie par l'expression suivante :

$$\forall v_i \in Dom(P_i), \; ! \exists v_j \in Dom(P_j) \wedge p_j \in Param^D(p_i) \Rightarrow v_j \in IParam^D(v_i)$$

4.2. Contraintes sémantiques

Cette section se focalise sur l'intégration des contraintes sémantiques dans le schéma dimensionnel. Ces contraintes agissent sur les processus d'implantation, d'interrogation et de manipulation des données. Les contraintes sémantiques sont liées au contexte d'analyse. Dans ce cadre, nous identifions des besoins non pris en compte dans les modèles actuels :

- un premier besoin concerne l'expression des relations entre les différentes hiérarchies d'un même axe d'analyse (dimension). Ces relations peuvent exprimer des interdictions entre les données des différentes hiérarchies. Ainsi, les instances relatives à une hiérarchie décrivant la géographie française ne peuvent être décrites suivant une hiérarchie relative à la géographie américaine. Ceci peut être traduit par des contraintes intra-dimensions portant sur les hiérarchies d'une même dimension ;
- un second besoin est relatif à la possibilité de spécifier selon quels axes (dimensions) et/ou quelles perspectives (hiérarchies) peuvent être associées les mesures d'activité (mesures d'un fait). Ces spécifications caractérisent les relations entre les hiérarchies, de différentes dimensions, qui peuvent être en conflit. Par exemple, les locations réalisées par des agences françaises (organisées selon la hiérarchies de la géographie française) sont analysées selon la hiérarchie de la classification française des véhicules et non pas avec la hiérarchie de la classification américaine. Cet aspect est pris en

compte par des contraintes sur les hiérarchies de dimensions distinctes. Nous appelons ces contraintes les contraintes inter-dimensions.

Dans les sections suivantes, nous présentons ces deux types de contraintes que nous avons intégrés dans notre modèle.

4.2.1. Contraintes intra-dimensions

Les contraintes intra-dimensions sont exprimées entre les hiérarchies d'une même dimension. Il s'agit de contraintes portant sur les instances de la dimension vérifiant la condition d'appartenance associée aux hiérarchies. A partir des relations entre les instances des hiérarchies nous définissons cinq contraintes intra-dimensions : l'exclusion, l'inclusion, la simultanéité, la partition et la totalité.

On pose D une dimension et $h_1 \in H^D$, $h_2 \in H^D$ deux hiérarchies définies sur D.

4.2.1.1. Exclusion intra-dimension

L'exclusion, notée $\otimes$, entre deux hiérarchies d'une dimension traduit qu'une instance de la dimension appartenant à une hiérarchie n'appartient pas à la seconde hiérarchie et réciproquement (une instance de la dimension appartient à une hiérarchie si elle satisfait la condition d'appartenance associée à la hiérarchie).

$$h_1 \otimes h_2 \text{ ssi } \forall I^D_{k1} \in_{(cond)} h_1 \wedge \forall I^D_{k2} \in_{(cond)} h_2 \Rightarrow I^D_{k1} \neq I^D_{k2}$$

Formalisme graphique. Nous adoptons le formalisme graphique suivant **:**

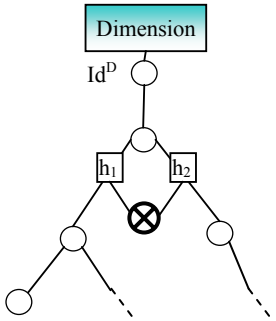


***Figure II.8** : Formalisme graphique de la contrainte d'exclusion intra-dimension.*

Exemple 5

Nous souhaitons exprimer le fait qu'une agence ne peut pas se situer en France et aux Etats-Unis en même temps (voir exemple Figure II.6). Ce besoin est exprimé par une contrainte d'exclusion entre les hiérarchies "*geo_us*" et "*geo_fr*" de la dimension *Agences*.

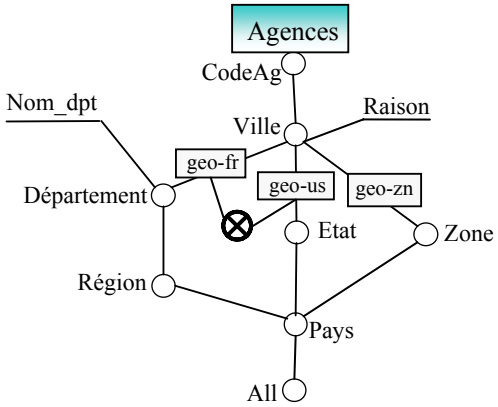


***Figure II.9** : Exemple de contrainte d'exclusion intra-dimension*

La Figure II.9 présente une illustration graphique d'une contrainte d'exclusion entre les hiérarchies "*geo_fr*" et "*geo_us*" de la dimension *Agences*. Dans la Figure II.10, nous visualisons les instances de la dimension *Agences*. Nous remarquons l'existence de deux ensembles disjoints représentant les agences françaises et les agences américaines. Ainsi, l'ensemble des agences $\{I^{AG}_3, I^{AG}_5\}$ appartenant à la hiérarchie "*geo_us*" est en exclusion avec l'ensemble des agences $\{I^{AG}_1, I^{AG}_2, I^{AG}_4\}$ appartenant à la hiérarchie "*geo_fr*".

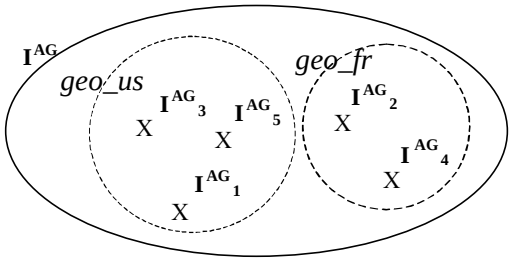


***Figure II.10** : Instances de la dimension sous contrainte d'exclusion intra-dimension*

4.2.1.2. Inclusion intra-dimension

L'inclusion, notée ⊙, entre deux hiérarchies d'une dimension traduit que toutes les instances de la dimension appartenant à une première hiérarchie appartiennent à la seconde hiérarchie.

$$h_1 \odot h_2 \text{ ssi } \forall I^D_k \in_{(cond)} h_1 \Rightarrow I^D_k \in_{(cond)} h_2$$

Formalisme graphique. Nous adoptons le formalisme graphique suivant **:**

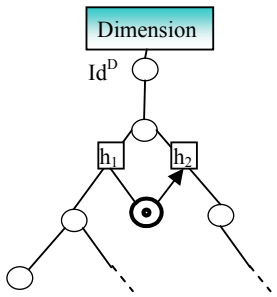


***Figure II.11** : Formalisme graphique de la contrainte d'inclusion intra-dimension.*

La flèche dans la figure indique le sens de l'inclusion de h_1 vers h_2.

Exemple 6

L'inclusion de la hiérarchie "*geo_fr*" dans "*geo_zn*", indique que toutes les instances vérifiant la condition d'appartenance à "*geo_fr*" sont impérativement des instances de la hiérarchie "*geo_zn*". Autrement dit, toutes les agences françaises appartiennent à une zone géographique.

La Figure II.12 présente graphiquement la contrainte d'inclusion entre les hiérarchies "*geo_fr*" et "*geo_zn*" par le symbole ⊙.

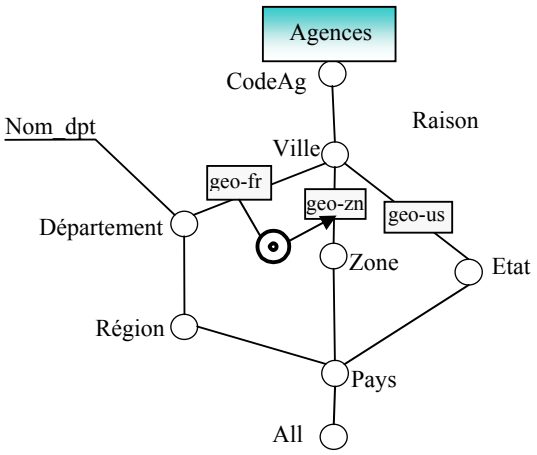


***Figure II.12** : Exemple de contrainte d'inclusion intra-dimension*

Dans la Figure II.13, nous visualisons les instances de la dimension *Agences*. Nous remarquons l'existence de deux ensembles en inclusion représentant les instances de la hiérarchie *"geo_fr"* et les instances de la hiérarchie *"geo_zn"*.

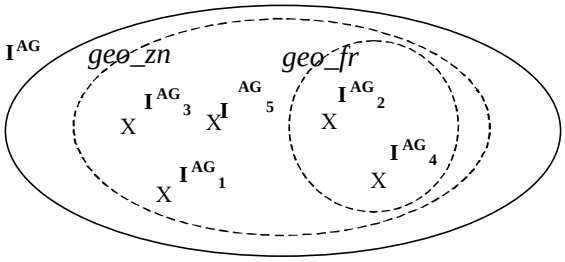


***Figure II.13** : Instances de la dimension sous contrainte d'inclusion intra-dimension*

4.2.1.3. Simultanéité intra-dimension

La simultanéité, notée ⊖, entre deux hiérarchies d'une dimension traduit que toutes les instances de la dimension appartenant à une première hiérarchie appartiennent à la seconde et réciproquement. Ainsi, toute instance de la dimension appartenant à l'une des deux hiérarchies, appartient alors également à l'autre.

$$h_1 \ominus h_2 \text{ ssi } \forall I^D_k \in_{(cond)} h_1 \Leftrightarrow I^D_k \in_{(cond)} h_2.$$

Remarque : $h_1 \ominus h_2 \Leftrightarrow h_1 \odot h_2 \wedge h_2 \odot h_1$.

Formalisme graphique. Nous adoptons le formalisme graphique suivant **:**

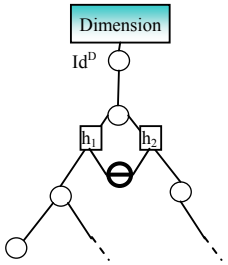


***Figure II.14** : Contrainte de simultanéité intra-dimension.*

Exemple 7

Nous souhaitons exprimer le fait que tous les employés doivent être classifiés indifféremment selon leur âge ou leur sexe. Ce besoin est exprimé par une contrainte de simultanéité définie sur les hiérarchies "*emp_age*" et "*emp_sex*" de la dimension *Employés*. Cette contrainte implique que toutes les instances de la première hiérarchie appartiennent à la deuxième hiérarchie et inversement.

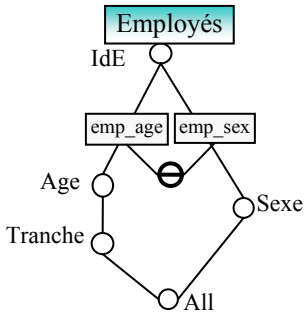


***Figure II.15** : Exemple de contrainte de simultanéité intra-dimension*

La Figure II.15 illustre graphiquement la contrainte de simultanéité entre les hiérarchies "*emp_age*" et "*emp_sex*" de la dimension *Employés*. Dans la Figure II.16, nous visualisons les instances de ces deux hiérarchies. Ces instances vérifient la contrainte de simultanéité entre les deux hiérarchies et représentent le même ensemble d'instances.

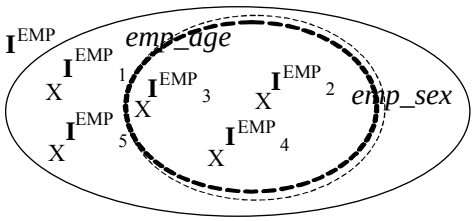


***Figure II.16** : Instances de la dimension sous contrainte de simultanéité intra-dimension*

4.2.1.4. Totalité intra-dimension

La totalité, notée ⊖, entre deux hiérarchies d'une dimension traduit le fait que toute instance de la dimension appartient à l'une ou (non exclusif) l'autre des hiérarchies. Ainsi, toute instance de la dimension appartient à l'une des deux hiérarchies et éventuellement aux deux hiérarchies.

$$h_1 \ominus h_2 \text{ ssi } \forall I^D{}_k \in I^D, I^D{}_k \in_{(cond)} h_1 \vee I^D{}_k \in_{(cond)} h_2.$$

Formalisme graphique. Nous adoptons le formalisme graphique suivant **:**

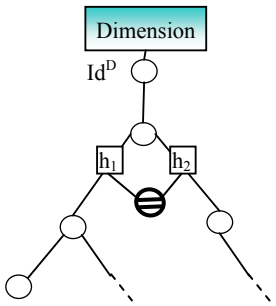


***Figure II.17** : Formalisme graphique de la contrainte de totalité intra-dimension.*

Exemple 8

Nous souhaitons exprimer le fait que les clients de nos agences sont de deux types : publique ou privé, sachant qu'un client peut être de type semi-publique (publique et privé à la fois). Ce besoin est exprimé par une contrainte de totalité entre les hiérarchies "*clt_Pub*" et "*clt_Privé*", qui implique que l'union des instances de ces deux hiérarchies forme la totalité des instances de la dimension *Clients*.

Nous définissons la contrainte de totalité afin de caractériser les hiérarchies qui contiennent toutes les instances de la dimension. Ainsi, si nous souhaitons calculer la somme des locations pour tous les clients (All), nous devons réaliser la somme des montants agrégés par *Ministère* et ceux agrégés par *Type*. C'est la contrainte de totalité entre les hiérarchies "*clt_Pub*" et "*clt_Privé*" qui permet d'assurer que le résultat trouvé est égal à la somme des montants des locations de tous les clients.

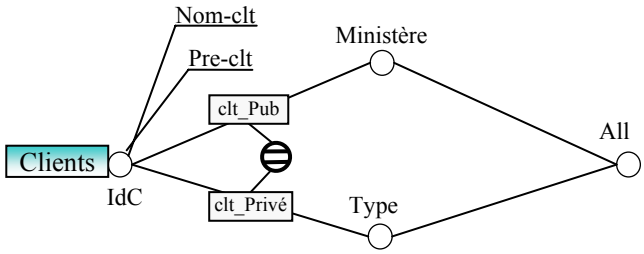


***Figure II.18** : Exemple de contrainte de totalité intra-dimension*

La Figure II.18 est la représentation graphique de la contrainte de totalité entre les hiérarchies "*clt_Pub*" et "*clt_Privé*" de la dimension *Clients*. Dans la Figure II.19, nous visualisons les instances de la dimension *Clients* réparties entre les deux hiérarchies pouvant contenir des éléments communs. Ainsi, les instances $\{I^{CLT}_1, I^{CLT}_3\}$ représentent des clients privés, les instances $\{I^{CLT}_2, I^{CLT}_4\}$ sont des clients publiques alors que l'instance $\{I^{CLT}_5\}$ représente un client semi_publique.

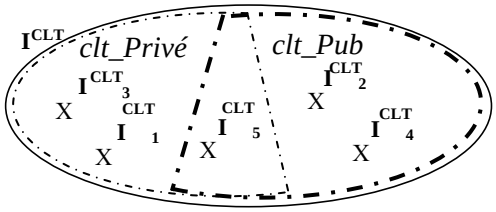


***Figure II.19** : Instances de la dimension sous contrainte de totalité intra-dimension*

4.2.1.5. Partition intra-dimension

La partition, notée ⊘, entre deux hiérarchies d'une dimension traduit que toute instance de la dimension appartient à l'une ou (exclusif) l'autre des hiérarchies. Ainsi, toute instance de la dimension appartient à l'une des deux hiérarchies, mais pas aux deux, ni à aucune des deux.

$$h_1 \oslash h_2 \text{ ssi}$$
$$(\forall I^D_k \in I^D, I^D_k \in_{(cond)} h_1 \veebar I^D_k \in_{(cond)} h_2)$$
$$\wedge (\forall I^D_{k1} \in_{(cond)} h_1 \wedge \forall I^D_{k2} \in_{(cond)} h_2 \Rightarrow I^D_{k1} \neq I^D_{k2}).$$

Remarque : $h_1 \oslash h_2 \Leftrightarrow h_1 \ominus h_2 \wedge h_1 \otimes h_2$.

Formalisme graphique. Nous adoptons le formalisme graphique suivant **:**

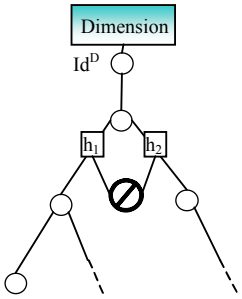


***Figure II.20** : Formalisme graphique de la partition intra-dimension.*

Exemple 9

Nous souhaitons exprimer le fait qu'il existe deux organisations possibles pour toutes les agences dans notre application décisionnelle : organisation géographique française ou américaine. Ce besoin est exprimé à l'aide d'une contrainte de partition entre les hiérarchies "*geo_us*" et "*geo_fr*", qui implique que l'intersection des instances des deux hiérarchies est vide (exclusion) et que l'union de ces instances forme la totalité des instances de la dimension *Agences* (totalité).

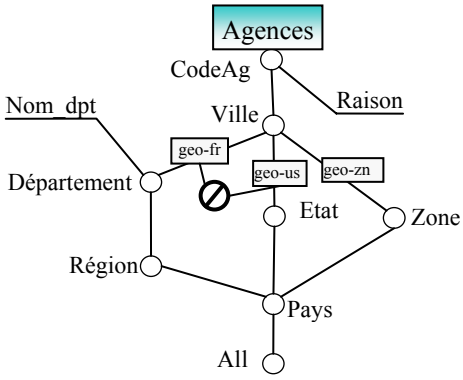


***Figure II.21** : Exemple de contrainte de partition intra-dimension*

La Figure II.21 présente une illustration graphique d'une contrainte de partition entre les hiérarchies "*geo_fr*" et "*geo_us*" de la dimension *Agences*. Dans la Figure II.22, nous

visualisons les instances de la dimension *Agences* réparties sur deux ensembles disjoints représentants les instances des hiérarchies *"geo_fr"* et *"geo_us"*.

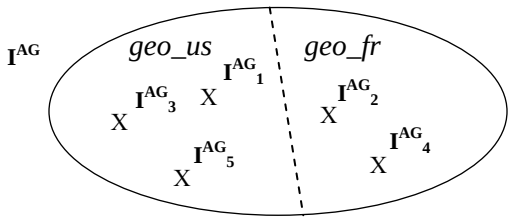


***Figure II.22** : Instances de la dimension sous contrainte de partition intra-dimension*

4.2.2. Contraintes inter-dimensions

Les contraintes inter-dimensions sont exprimées entre les hiérarchies de dimensions distinctes. Ces contraintes décrivent les relations entre les données d'un fait en considérant les perspectives d'analyse appliquées. Il s'agit de contraintes portant sur les instances du fait associées aux sous-ensembles d'instances des hiérarchies des dimensions.

On pose D_1 et D_2 deux dimensions associées à un fait F ($D_1 \in Star^C(F) \wedge D_2 \in Star^C(F)$) et $h_1 \in H^{D1}$, $h_2 \in H^{D2}$ deux hiérarchies des dimensions D_1 et D_2.

4.2.2.1. Exclusion inter-dimensions

L'exclusion entre deux hiérarchies h_1 et h_2 de dimensions distinctes, respectivement D_1 et D_2, traduit qu'une instance d'un fait liée à une instance de la dimension D_1 appartenant à la hiérarchie h_1 n'est pas associée à une instance de la dimension D_2 appartenant à la hiérarchie h_2 et réciproquement.

$$h_1 \otimes h_2 \text{ ssi}$$
$$(\forall I^F{}_j \in I^F \mid \exists I^{D1}{}_{k1} \in_{(cond)} h_1 \wedge I^{D1}{}_{k1} \in IStar^F(I^F{}_j))$$
$$\Rightarrow \neg (\exists I^{D2}{}_{k2} \in_{(cond)} h_2 \mid I^{D2}{}_{k2} \in Istar^F(I^F{}_j)).$$

Formalisme graphique. Nous adoptons le formalisme graphique suivant **:**

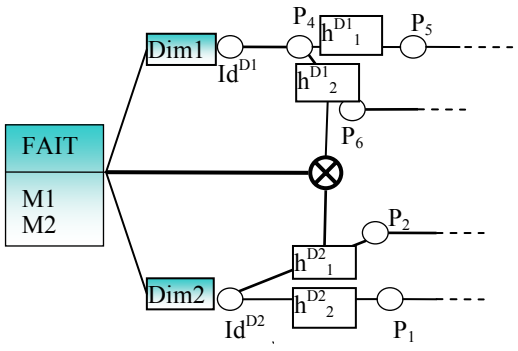


***Figure II.23** : Formalisme graphique de la contrainte d'exclusion inter-dimensions.*

Exemple 10

On considère la dimension *Véhicules* comportant les hiérarchies suivantes :

- $h^{Véhicules}_1$ = {"clas_us", {$Param^{clas\_us}$(Immat) = Type_Mot, $Param^{clas\_us}$(Type_Mot) = Class, $Param^{clas\_us}$(Class) = All}, Class ≠ NULL ∧ TypeMot ≠ NULL),
- $h^{Véhicules}_2$ = {"clas_veh", {$Param^{clas\_us}$(Immat) = Genre, $Param^{clas\_us}$(Genre) = Vitesse, $Param^{clas\_us}$(Vitesse) = All}, TRUE),
- $h^{Véhicules}_3$ = {"clas_fr", {$Param^{clas\_us}$(Immat) = Modèle, $Param^{clas\_us}$(Modèle) = Marque, $Param^{clas\_us}$(Marque) = All}, Marque ≠ NULL).

La hiérarchie "*clas_us*", spécifique aux Etats-Unis, décrit les véhicules suivant une classification en type de moteur, puis en classe de véhicule (confort, luxe, …), tandis que la hiérarchie "*clas_fr"* décrit une nomenclature inhérente à la France. La hiérarchie "*clas_veh*" organise les véhicules à louer selon leur genre (sport, citadine, …) et en nombre de vitesse.

Le concepteur souhaite exprimer le fait que les locations de véhicules classées selon la nomenclature des Etats-Unis ne peuvent pas être analysées suivant les agences françaises (ces dernières ne les proposent pas à la location). Ce besoin est exprimé à l'aide d'une contrainte d'exclusion entre les hiérarchies "*clas_us*" et "*clas_fr*", qui implique que les instances du fait *Location* reliées à la première hiérarchie ne sont pas en relation avec les instances de la hiérarchie "*geo_fr*".

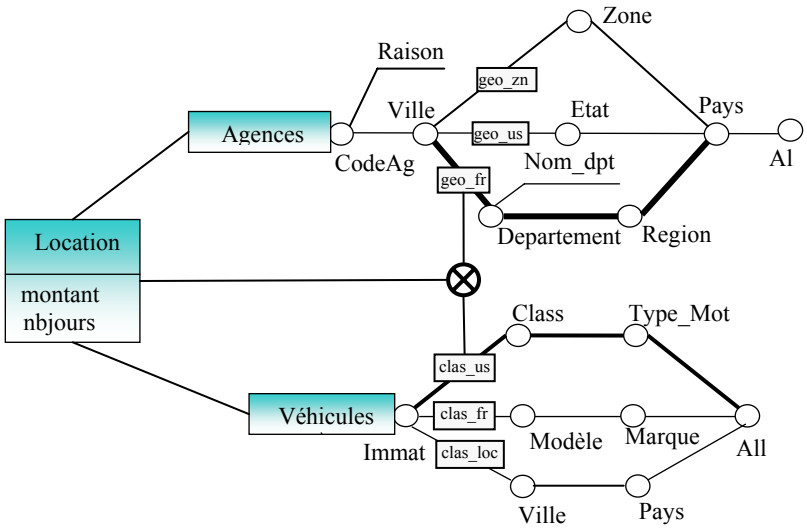


***Figure II.24** : Exemple de contrainte d'exclusion inter-dimensions*

La Figure II.24 présente une contrainte d'exclusion inter-dimensions entre les hiérarchies "*geo_fr*" de la dimension *Agences* et la hiérarchie "*clas_us*" de la dimension *Véhicules*.

Dans la Figure II.25, nous visualisons les instances du fait *Location* qui forment deux ensembles disjoints. Le premier ensemble (①) représente les instances du fait *Location* qui sont associées aux instances de la dimension *Véhicules* organisées selon la hiérarchie "*clas_us*". Le deuxième ensemble (②) représente les instances du fait *Location* reliées aux agences françaises (instances de la hiérarchie "*geo_fr*"). Les deux ensembles ① et ② vérifient la contrainte d'exclusion entre les deux hiérarchies citées précédemment.

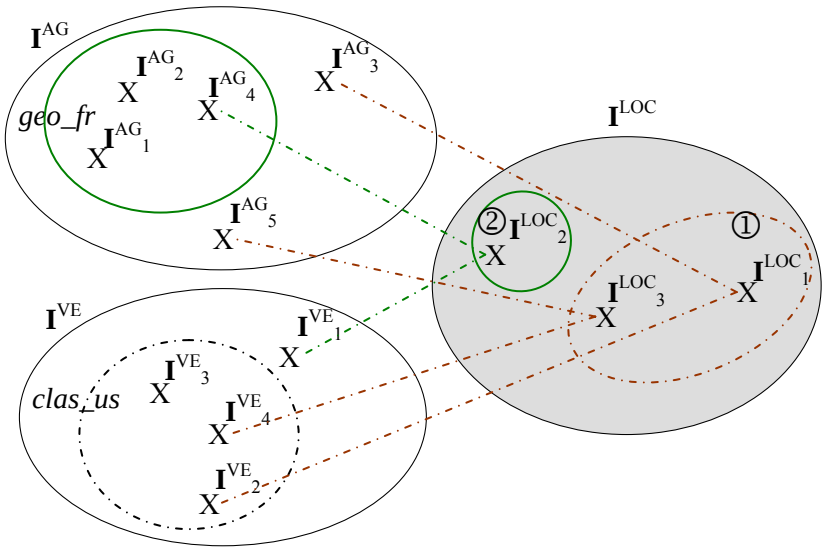


***Figure II.25** : Instances du fait et des dimensions sous une contrainte d'exclusion inter-dimensions.*

4.2.2.2. Inclusion inter-dimensions

L'inclusion entre deux hiérarchies de dimensions distinctes traduit que toutes les instances d'un fait liées aux instances de la dimension appartenant à une première hiérarchie sont également liées aux instances appartenant à la seconde hiérarchie.

$$h_1 \circledcirc h_2 \text{ ssi}$$
$$(\forall I^F_j \in I^F \mid \exists I^{D1}_{k1} \in_{(cond)} h_1 \wedge I^{D1}_{k1} \in Istar^F(I^F_j))$$
$$\Rightarrow (\exists I^{D2}_{k2} \in_{(cond)} h_2 \mid I^{D2}_{k2} \in IStar^F(I^F_j))$$

Formalisme graphique. Nous adoptons le formalisme graphique suivant **:**

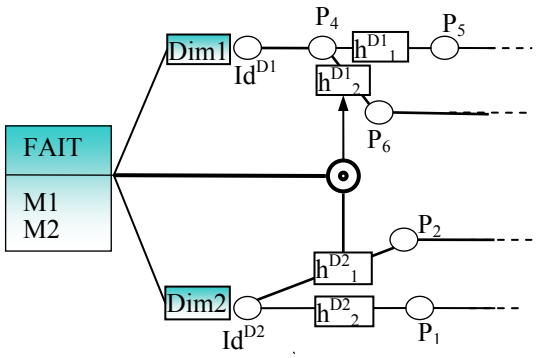


***Figure II.26** : Formalisme graphique de la contrainte d'inclusion inter-dimensions.*

La flèche dans la figure indique le sens de l'inclusion de $h^{D2}{}_1$ vers $h^{D1}{}_2$.

Exemple 11

Nous souhaitons exprimer le fait que toutes les locations de véhicules classés selon la nomenclature des Etats-Unis, doivent être analysées suivant les zones géographiques. Ce besoin est exprimé à l'aide de la contrainte d'inclusion de la hiérarchie "*clas_us*"

dans la hiérarchie "*geo_zn*", qui implique que les instances du fait Location reliées à la première hiérarchie sont toutes en relation avec les instances de la hiérarchie "*geo_zn*".

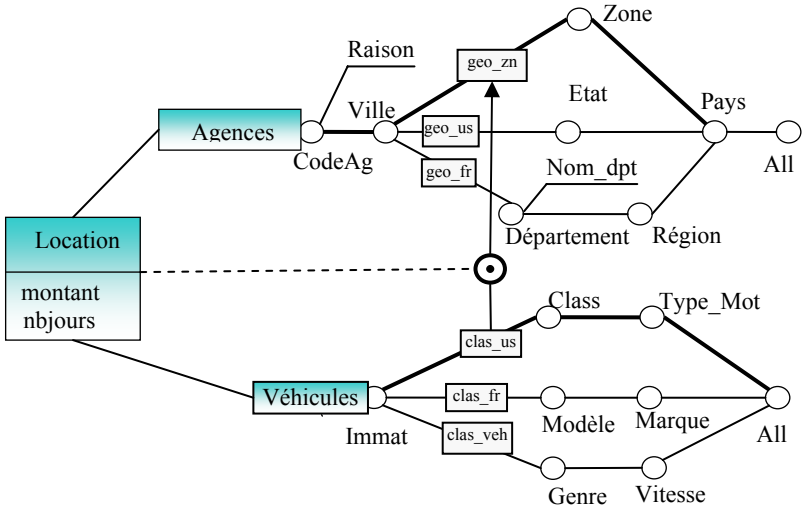


***Figure II.27** : Exemple de contrainte d'inclusion inter-dimensions*

La Figure II.27 présente une contrainte d'inclusion inter-dimensions de la hiérarchie "*clas_us*" de la dimension *Véhicules* dans la hiérarchie "*geo_zn*" de la dimension *Agences.*

Dans la Figure II.28, nous visualisons les instances du fait *Location* comportant un sous ensemble ① qui représente les instances reliées aux véhicules organisés selon la nomenclature américaine. Les instances de ce sous ensemble appartiennent à l'ensemble des locations reliées aux instances de la hiérarchie "*geo_zn*" (②). Les instances du fait *Location* vérifient la contrainte d'inclusion de la hiérarchie "*clas_us*" dans "*geo_zn*".

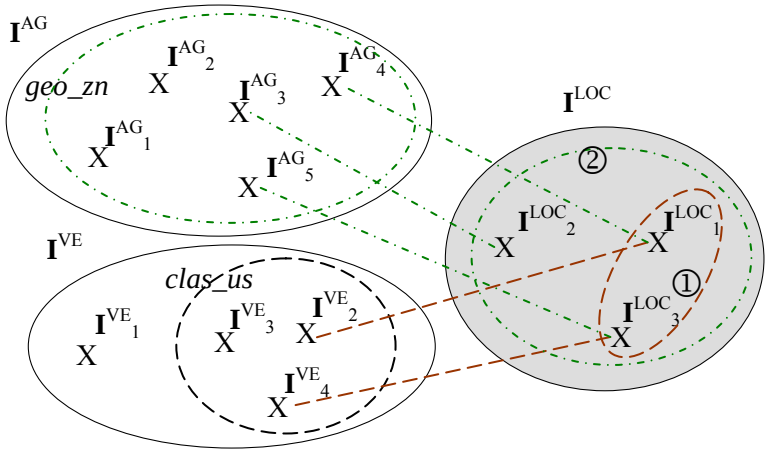


***Figure II.28** : Instances du fait et des dimensions sous une contrainte d'inclusion inter-dimensions.*

4.2.2.3. Simultanéité inter-dimensions

La simultanéité entre deux hiérarchies de dimensions distinctes traduit que toutes les instances d'un fait liées aux instances de la dimension, appartenant à une première hiérarchie, sont également liées aux instances appartenant à la seconde et réciproquement.

$$h_1 \ominus h_2 \text{ ssi}$$
$$(\forall I^F_j \in I^F \mid \exists I^{D1}_{k1} \in_{(cond)} h_1 \wedge I^{D1}_{k1} \in IStar^F(I^F_j))$$
$$\Leftrightarrow \exists I^{D2}_{k2} \in_{(cond)} h_2 \mid I^{D2}_{k2} \in IStar^F(I^F_j)$$

Remarque : $h_1 \ominus h_2 \Leftrightarrow h_1 \circledcirc h_2 \wedge h_2 \circledcirc h_1$.

Formalisme graphique. Nous adoptons le formalisme graphique suivant **:**

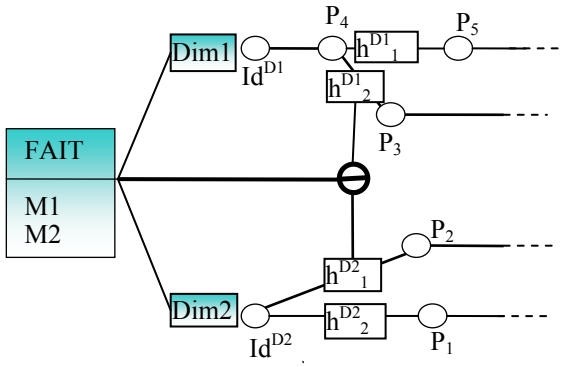


***Figure II.29** : Formalisme graphique de la contrainte de simultanéité inter-dimensions*

Exemple 12

Nous souhaitons exprimer le fait qu'une location de véhicules réalisée dans une agence américaine est nécessairement analysée selon la classification américaine des véhicules. Ce besoin est exprimé par la contrainte de simultanéité entre les hiérarchies "*clas_us*" et "*geo_us*", qui implique que les instances du fait *Location* reliées aux instances de la hiérarchie américaine des véhicules sont simultanément reliées aux instances de la hiérarchie décrivant les agences suivant la géographique américaine.

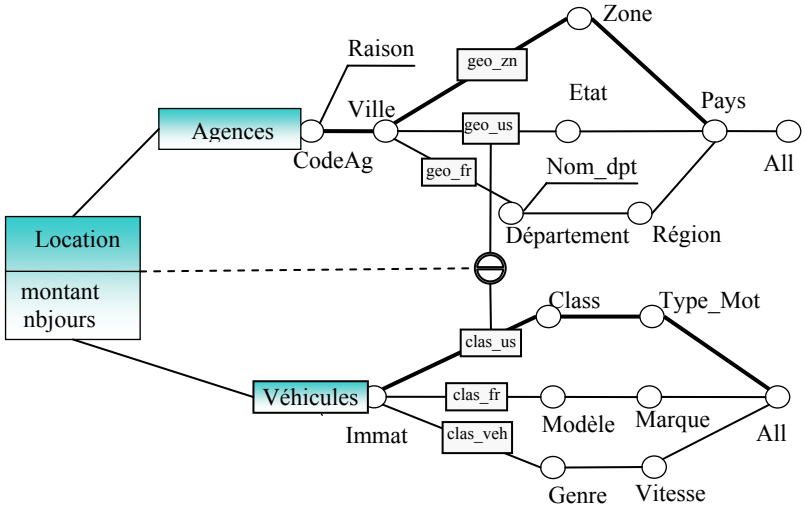


***Figure II.30** : Exemple de contrainte de simultanéité inter-dimensions*

La Figure II.30 présente une contrainte de simultanéité inter-dimensions entre les hiérarchies "*clas_us*" de la dimension *Véhicules* et "*geo_us*" de la dimension *Agences.*

Dans la Figure II.31, nous visualisons les instances du fait *Location* (ensemble ①) reliées en même temps aux instances de la hiérarchie "*geo_us*" et aux instances de la

hiérarchie "*clas_us*". Les éléments de ce sous ensemble vérifient la contrainte de simultanéité entre les hiérarchies "*clas_us*" et "*geo_us*".

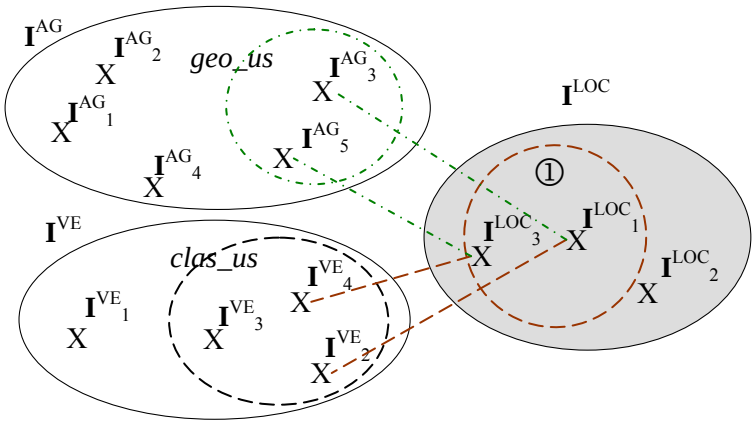


***Figure II.31** : Instances du fait et des dimensions sous une contrainte de simultanéité inter-dimensions*

4.2.2.4. Totalité inter-dimensions

La totalité entre deux hiérarchies de dimensions distinctes traduit le fait que toute instance du fait est liée à une instance appartenant à l'une des deux hiérarchies et éventuellement aux deux hiérarchies.

$$h_1 \ominus h_2 \text{ ssi}$$
$$\forall I^F_j \in I^F, (\exists I^{D1}_{k1} \in_{(cond)} h_1 \mid I^{D1}_{k1} \in Istar^F(I^F_j)) \vee$$
$$(\exists I^{D2}_{k2} \in_{(cond)} h_2 \mid I^{D2}_{k2} \in IStar^F(I^F_j))$$

Formalisme graphique. Nous adoptons le formalisme graphique suivant **:**

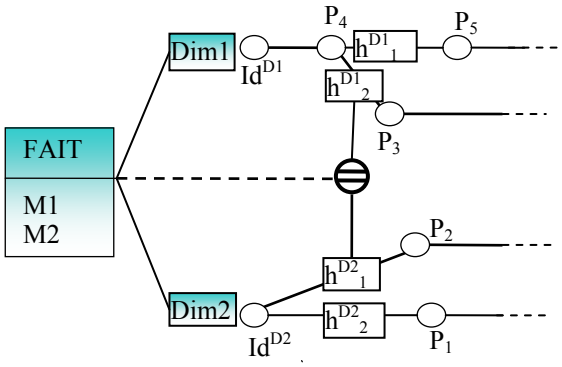


***Figure II.32** : Formalisme graphique de la contrainte de totalité inter-dimensions.*

Exemple 13

Nous désirons exprimer que toutes les locations sont relatives aux Etats-Unis ou à la France ; plus précisément, toutes les locations réalisées concernent soit des véhicules classés selon la nomenclature des Etats-Unis, soit des agences françaises. Ce fait est représenté par une contrainte de totalité entre la hiérarchie "*clas_us*" et la hiérarchie "*geo_fr*", qui implique que l'union des instances du fait Location reliées à la première

hiérarchie et celles reliées à la deuxième hiérarchie représente la totalité des instances du fait.

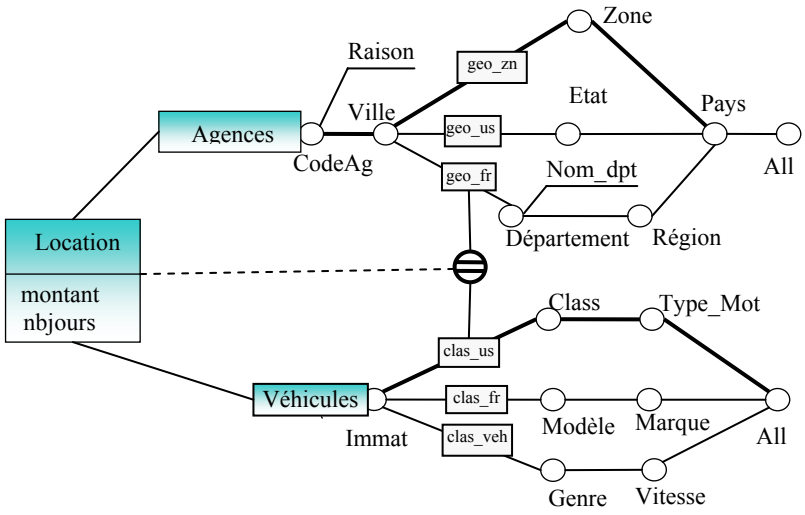


***Figure II.33** : Exemple de contrainte de totalité inter-dimensions*

La Figure II.33 présente graphiquement la contrainte de totalité entre les hiérarchies "geo_fr" de la dimension *Agences* et "*clas_us*" de la dimension *Véhicules* par le symbole ⊖.

La Figure II.34 est une illustration d'une contrainte de totalité. Nous visualisons l'ensemble des instances du fait *Location* associées soit aux instances de la hiérarchie de la nomenclature américaine des véhicules "*clas_us*", soit aux instances de la hiérarchie géographie française "*geo_fr*". En effet, il n'existe pas de location qui n'est liée ni à l'une ni à l'autre de ces deux hiérarchies.

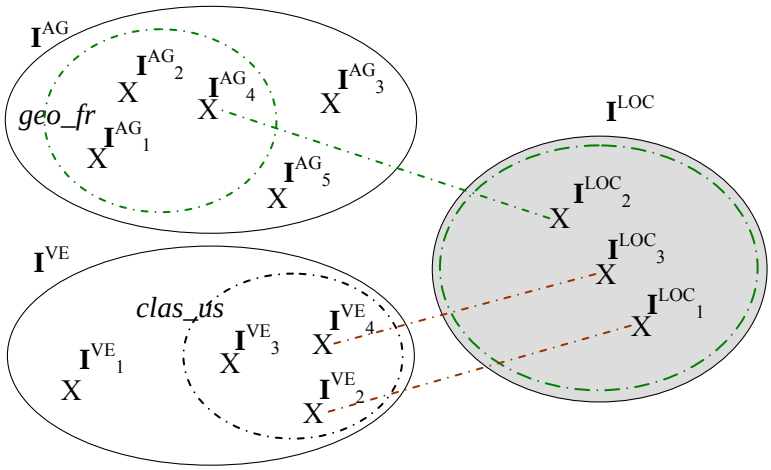


***Figure II.34** : Instances du fait et des dimensions sous une contrainte de totalité inter-dimensions*

4.2.2.5. Partition inter-dimensions

La partition entre deux hiérarchies h_1 et h_2 indique que chaque instance du fait est associée soit aux instances de h_1 soit à celles de h_2 (ou exclusif).

$$h_1 \oslash h_2 \text{ ssi } (\forall I^F{}_j \in I^F, (\exists I^{D1}{}_{k1} \in_{(cond)} h_1 \mid I^{D1}{}_{k1} \in IStar^F(I^F{}_j))$$
$$\vee (\exists I^{D2}{}_{k2} \in_{(cond)} h_2 \mid I^{D2}{}_{k2} \in Istar^F(I^F{}_j))) \wedge$$
$$((\forall I^F{}_j \in I^F \mid \exists I^{D1}{}_{k1} \in_{(cond)} h_1 \wedge I^{D1}{}_{k1} \in IStar^F(I^F{}_j))$$
$$\Rightarrow \neg(\exists I^{D2}{}_{k2} \in_{(cond)} h_2 \mid I^{D2}{}_{k2} \in IStar^F(I^F{}_j))).$$

Remarque : $h_1 \oslash h_2 \Leftrightarrow h_1 \ominus h_2 \wedge h_1 \otimes h_2$.

Formalisme graphique. Nous adoptons le formalisme graphique suivant **:**

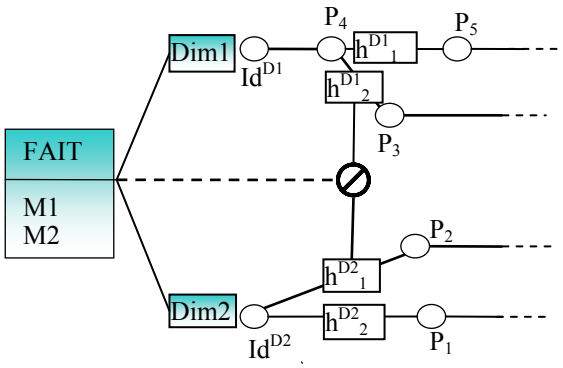


***Figure II.35** : Formalisme graphique de la contrainte de partition inter-dimensions*

Exemple 14

Entre les hiérarchies "*clas_us*" et "*geo_fr*", nous avons défini deux contraintes : exclusion et totalité. Ces deux contraintes réunies expriment une contrainte de partition.

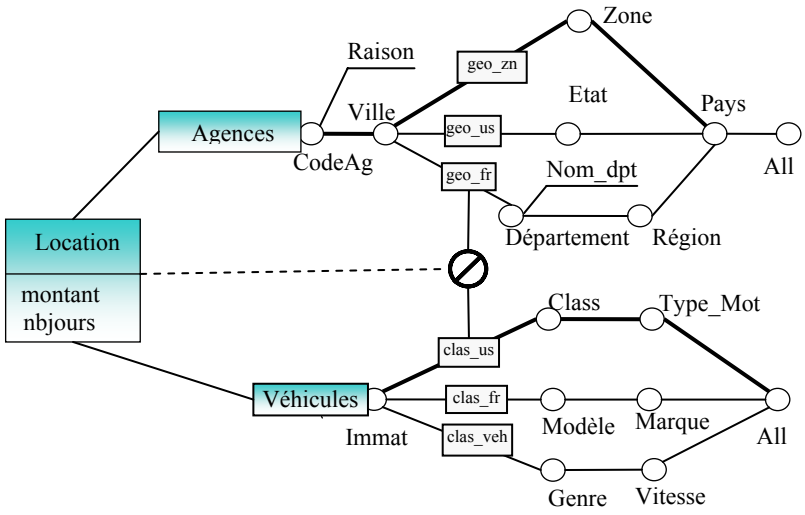


***Figure II.36** : Exemple de contrainte de partition inter-dimensions*

La Figure II.36 présente une contrainte de partition inter-dimensions entre les hiérarchies "*clas_us*" de la dimension *Véhicules* et "*geo_fr*" de la dimension *Agences.*

Dans la Figure II.37, nous visualisons les instances du fait *Location* réparties entre deux ensembles disjoints. Un ensemble représentant les locations analysées en fonction des véhicules organisés selon la nomenclature américaine. Un deuxième ensemble de locations reliées aux instances de la hiérarchie "*geo_fr*".

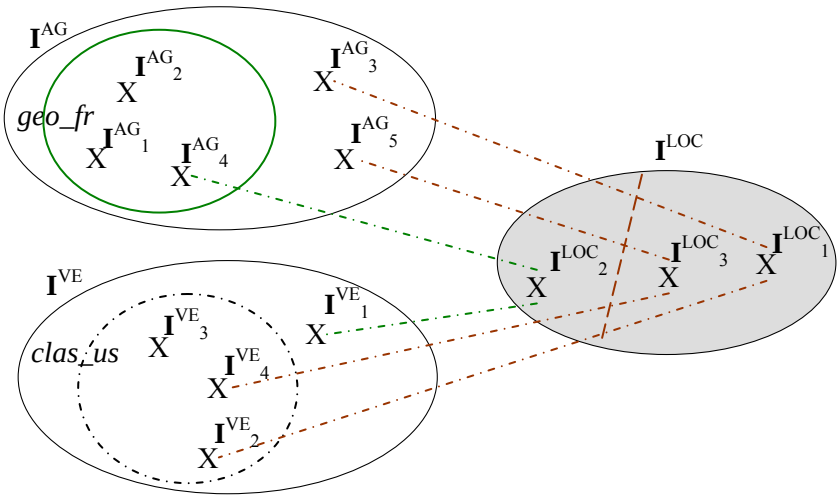


***Figure II.37** : Instances du fait et des dimensions sous une contrainte de partition inter-dimensions*

5. Conclusion

Dans ce chapitre, nous avons présenté notre modèle conceptuel pour les données dimensionnelles. Ce modèle est basé sur le concept de constellation regroupant un ensemble de faits analysés suivant des dimensions ou axes d'étude (Ghozzi *et al*, 2003a) (Ghozzi *et al*, 2003c). Le modèle en constellation permet de répondre à un premier besoin des bases de données dimensionnelles en facilitant la corrélation entre les différents sujets d'analyse partageant des dimensions. Ces dernières sont organisées en hiérarchies multiples permettant l'analyse multi-perspective d'un même axe d'analyse. Les hiérarchies sont définies explicitement dans notre modèle afin de faciliter la manipulation des données dimensionnelles en fonction des différents niveaux de granularités. Ces hiérarchies intègrent également l'expression des attributs faibles. Nous proposons conjointement à chaque concept un formalisme graphique visant à simplifier la représentation du schéma dimensionnel.

Dans notre modèle, nous proposons de gérer le temps d'une manière spécifique en ne gardant que l'information pertinente à l'analyse. Nous archivons les données après les avoir agrégées en enlevant les détails devenus obsolètes. Ainsi, la gestion des données historisées est réalisée à deux niveaux : détaillé et archivé. Cette gestion est assurée par la définition de deux hiérarchies dans la dimension *Temps* en spécifiant pour chacune d'elles l'ensemble des données analysées : la hiérarchie "*T_Det*" décrit les données détaillées et la hiérarchie "*T_Arch*" décrit les données archivées.

Notre modèle assure une plus grande cohérence des données par :

- sa propriété de multi-instanciation (condition d'appartenance des instances des dimensions aux hiérarchies) ;
- un ensemble de contraintes structurelles et sémantiques. Ces contraintes offrent une représentation plus précise de la réalité et visent à interdire les corrélations incohérentes lors des analyses.

Les contraintes structurelles portent sur les concepts et les instances du modèle dimensionnel. Au niveau des concepts, les contraintes syntaxiques décrivent les règles

d'intégrité du modèle dimensionnel, telles que l'unicité des noms des concepts, tandis que les contraintes hiérarchiques caractérisent les relations entre les niveaux hiérarchiques au sein des dimensions. Au niveau des instances, nous avons défini des contraintes sur les liens des instances d'un fait et celles de ses dimensions d'une part, et les liens entre les instances des différents niveaux hiérarchiques, d'autre part.

Les contraintes sémantiques expriment l'inclusion, l'exclusion, la simultanéité, la totalité et la partition entre les hiérarchies (Ghozzi *et al*, 2003b). Nous distinguons les contraintes intra-dimension qui permettent de caractériser les relations entre les instances des hiérarchies d'une même dimension, des contraintes inter-dimensions qui caractérisent les interactions entre les instances des faits reliées aux instances des hiérarchies de dimensions distinctes. Ces contraintes permettent d'exprimer toutes les règles sémantiques (règles de gestion, économiques, géographiques, …) complémentaires au modèle conceptuel. La validation de ces règles permet d'assurer l'intégrité des données dimensionnelles.

La proposition d'un modèle conceptuel constitue la première étape de la conception de bases de données dimensionnelles à contraintes. L'étude de l'impact de ces contraintes sur les opérations dimensionnelles (cet aspect est peu étudié dans les travaux existants) est réalisée dans le chapitre III.

CHAPITRE III : INTERROGATION DE DONNEES DIMENSIONNELLES CONTRAINTES

PLAN DU CHAPITRE

La présentation des données dans le cadre des systèmes décisionnels répond davantage à des besoins de visualisation d'un grand volume de données à organiser de manière flexible en fonction des objectifs décisionnels. Les langages d'interrogation des données dimensionnelles ont été proposés pour répondre à ces besoins (Marcel, 1998). Notamment, ces langages intègrent :

- la structuration des données de manière à rapprocher les données et les traitements des décideurs, d'assurer l'intégrité des données et la fiabilité des résultats restitués et d'améliorer le confort de travail (en réduisant par exemple les temps de réponse) ;
- la prise en compte du point de vue du décideur lors de la définition de l'interface d'interrogation.

Ce chapitre se focalise sur l'interrogation des données dimensionnelles de manière cohérente et fiable. Dans la première section, nous présentons notre problématique et nos propositions dans ce cadre. La deuxième section présente notre langage d'interrogation de données dimensionnelles en intégrant un ensemble de contraintes sémantiques qui assurent la fiabilité des données. La troisième section décrit notre approche pour l'intégration des contraintes sémantiques dans le calcul des pré-agrégats (vues matérialisées). Le but de l'approche est d'optimiser l'interrogation des données dimensionnelles.

1. Introduction à l'interrogation des données dimensionnelles

La constitution d'une base de données décisionnelle fiable et cohérente implique l'intégration de contraintes dans un modèle dimensionnel (Hurtado *et al*. 2002). Les contraintes permettent une représentation plus précise de la réalité et interdisent des corrélations incohérentes, offrant ainsi un cadre plus fiable pour les analyses et les prises de décision. Afin de répondre à ce besoin, nous avons proposé dans le chapitre précédent, un modèle dimensionnel qui intègre un ensemble de contraintes sémantiques (Ghozzi et al, 2003b). L'intégration de ces contraintes dans le modèle dimensionnel est une première étape dans la constitution d'une base dimensionnelle cohérente. La seconde étape consiste en l'exploitation de ces contraintes lors de la manipulation « *optimisée* » d'une base de données dimensionnelles.

1.1. Problématique

En étudiant l'impact des contraintes sur les langages d'interrogation des données dimensionnelles, nous avons identifié deux champs complémentaires d'exploitation de ces contraintes. Un champ relatif à l'interrogation des données dimensionnelles et un deuxième qui concerne l'optimisation des interrogations basée sur la technique de matérialisation des vues.

Parallèlement aux modèles dimensionnels, plusieurs langages d'interrogation des données dimensionnelles ont été proposés (Li et al, 1996) (Agrawal et al, 1997) (Vassiliadis, 1998) (Abelló et al, 2003). Ces langages visent à répondre aux besoins décisionnels en proposant des opérateurs interactifs d'analyse en ligne facilitant la navigation entre les données dimensionnelles (Abelló et al, 2003). Seulement, aucun standard n'a été unanimement accepté pour la formalisation de ces opérateurs. En outre, les langages proposés se basent sur des modèles dimensionnels qui ne supportent pas l'expression des contraintes sémantiques (Hurtado et al, 2002). Or, l'analyse des données dimensionnelles qui ne tient pas compte de ces contraintes peut engendrer des incohérences et peut altérer les résultats.

Considérons l'exemple de l'analyse des locations des véhicules, (cf. chapitre II, §2.3) en fonction des dimensions *Agences* et *Temps*. Au niveau de la dimension *Agences*, nous avons défini les hiérarchies "*geo_fr*", décrivant la géographie française, et "*geo_us*", spécifique aux villes américaines. Dans l'exemple de la Figure III.1, la visualisation des montants des locations par *Ville* et par *Année* fait apparaître des valeurs nulles pour le paramètre *Etat* de la dimension *Agences*. Ces valeurs, causées par le fait que les agences françaises ne sont pas localisées dans des états, peuvent être mal interprétées par le décideur. Il peut considérer que ces données sont manquantes et que la fiabilité des résultats obtenus est douteuse.

Location (montant)		*Agences*			
		Etat	NULL	NULL	Texas
		Ville	Toulouse	Lyon	Dallas
Temps	Année				
	2002		(80)	(120)	(200)
	2001		(120)	(100)	(150)
	2000		(100)	(50)	(220)

***Figure III.1** : Visualisation sans contraintes des locations en fonction des villes et des années*

Cette problématique est encore plus tangible lorsqu'il s'agit d'analyser les données en combinant différentes dimensions comportant des hiérarchies incompatibles. Reprenons l'exemple de l'analyse de la Figure III.1 et remplaçons la dimension *Temps* par la dimension *Véhicules* et le paramètre *Année* par le paramètre *Marque* caractérisant les véhicules français. Nous constatons que la visualisation des montants des locations en fonction des villes et des marques des véhicules fait apparaître des valeurs nulles au niveau des mesures (cf. Figure III.2). Dans ce cas, le décideur peut facilement interpréter que la ville de '*Dallas'* n'a pas réalisé de locations alors que c'est plutôt la règle de gestion qui stipule que les agences américaines n'utilisent pas la classification des véhicules françaises qui a induit ces valeurs nulles pour cette ville.

Location (montant)		*Agences*			
		Ville	Toulouse	Lyon	Dallas
Véhicules	Marque				
	M1		(30)	(70)	NULL
	M2		(60)	(30)	NULL
	M3		(40)	(40)	NULL
	M4		(120)	(80)	NULL

***Figure III.2** : Visualisation des locations en fonction des villes et des marques de véhicules*

Le deuxième volet, dans notre étude de l'impact des contraintes sur la manipulation des données dimensionnelles, concerne l'intégration de la sémantique des contraintes dans la sélection des vues à matérialiser. Cette intégration vise à optimiser l'interrogation en diminuant les temps de réponse aux requêtes des décideurs.

Dans un schéma dimensionnel, la structure des données nous donne une idée sur les requêtes des décideurs. Ces requêtes sont, généralement, formées de l'agrégation des mesures en fonction des combinaisons de paramètres des dimensions. La technique de matérialisation des vues consiste à pré-calculer et à stocker les vues correspondant à ces combinaisons de paramètres afin de diminuer le temps de réponse aux requêtes des décideurs. Seulement, le stockage de toutes les vues possibles s'avère très coûteux, surtout si ces vues nécessitent un coût de rafraîchissement, d'où le problème de sélection des vues à matérialiser. Plusieurs travaux se focalisent sur ce problème (Harinarayan *et al*, 1996) (Baralis *et al,* 1997)

(Theodoratos *et al*. 1999) (Kotidis *et al*, 2001) (Paraboschi et al, 2003). Ils se basent sur des modèles classiques n'intégrant pas de contraintes sémantiques. Cette limite entraîne des constructions de vues matérialisées inadéquates. Reprenons l'exemple de l'analyse des locations de véhicules en fonction des dimensions : *Agences, Temps* et *Véhicules*. Dans cet exemple, la combinaison des paramètres des dimensions *Agences* et *Véhicules* donnera lieu à la définition de vues matérialisées incohérentes telle que la vue combinant le paramètre *Marque* caractérisant les véhicules français et le paramètre *Etat* caractérisant une agence américaine. A notre connaissance, ce problème n'a été traité par aucun des travaux antérieurs.

1.2. Proposition

Dans ce chapitre, nous poursuivons l'étude des contraintes intégrées dans notre modèle dimensionnel en analysant leurs répercussions sur la manipulation et l'optimisation de l'interrogation des données dimensionnelles. Nous étudions leur impact à deux niveaux :

- la manipulation des données dimensionnelles en intégrant l'expression des contraintes dans les opérateurs dimensionnels afin d'assurer une manipulation cohérente des données. Cette intégration est réalisée, soit par une simple vérification des contraintes lors de la visualisation, soit par la proposition d'un langage d'interrogation dimensionnel permettant de tenir compte de la sémantique de ces contraintes ;
- le choix des vues à matérialiser afin d'améliorer le processus d'interrogation et diminuer les temps de réponses aux requêtes des décideurs. Ainsi, notre objectif dans la deuxième partie de ce chapitre est d'exploiter l'ensemble des contraintes sémantiques intégrées dans le modèle pour la sélection des vues matérialisées. Cette exploitation vise à éliminer des combinaisons incohérentes et ainsi à réduire le nombre de ces vues.

Dans les sections suivantes, nous proposons un langage d'interrogation dimensionnel intégrant les contraintes définies au niveau du modèle dimensionnel. Puis, nous présentons les répercussions des contraintes au niveau de la sélection des vues à matérialiser afin d'optimiser l'interrogation.

2. Langage d'interrogation des données dimensionnelles à contraintes

Les contraintes sémantiques intra et inter-dimensions agissent sur les opérations dimensionnelles (Ghozzi *et al,* 2003b). La prise en compte des contraintes nécessite la proposition d'un langage d'interrogation des données dimensionnelles :

- évitant des corrélations incohérentes,
- précisant l'ensemble des données à visualiser en fonction des besoins des décideurs.

Par la suite, nous utilisons la notation [ ] pour indiquer un paramètre optionnel dans la définition des opérateurs.

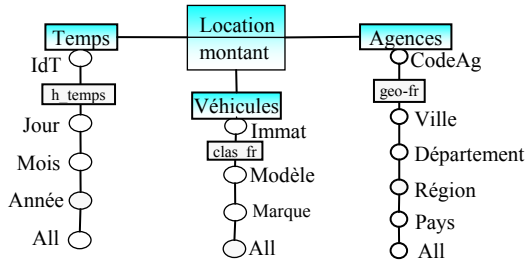
2.1. Préliminaire

Nous souhaitons proposer un concept permettant à un décideur (non informaticien) de visualiser de manière ergonomique les données d'un schéma en constellation. Nous définissons, pour cela, une **structure de table dimensionnelle** adaptée à notre schéma en constellation qui regroupe plusieurs faits et intègre des hiérarchies multiples. Ce concept facilite la manipulation des données dimensionnelles grâce à la simplicité de la représentation en tableau qui permet de visionner l'information de manière intuitive. Notre proposition est

une adaptation de la table N-dimensionnelle proposée par (Gyssen *et al*, 1997) (Voir Chapitre I § 2.2.1).

Exemple 1

Nous présentons dans la Figure III.3 un exemple de schéma dimensionnel et sa visualisation en table dimensionnelle. Ce schéma comporte le fait *Location* analysé suivant trois dimensions : *Temps*, *Véhicules* et *Agences*. Ce schéma est accompagné de sa représentation en table dimensionnelle. La table permet de visualiser les montants des locations par *Année* et par *Région* pour tous les véhicules (dimension *Véhicules* analysées en fonction du paramètre All). Un seul fait est représenté dans cette table puisque le schéma dimensionnel ne comporte que le fait *Location*.



Location (montant)		*Agences*	*(geo_fr)*	
		Pays	France	
		Régions	MP	AQ
Temps (h_temps)	Année			
	2002		80	120
	2001		100	50
Véhicules.All='All'				

Figure III.3 *: Représentation d'un schéma dimensionnel par une table dimensionnelle.*

♦ ***Définition de la Table dimensionnelle***

Définition

Une table dimensionnelle permet de visualiser les données sous forme de lignes, de colonnes et de plans. Les deux dimensions, visualisées dans le plan en cours, sont affichées sur les lignes et colonnes. Les mesures du fait visualisé sont placées à l'intersection d'une ligne et d'une colonne. La liste des autres faits est disponible à droite de la table dimensionnelle (cf. Figure III.4).

Une table dimensionnelle, notée TD^{Fc}*, est définie comme suit :*

$TD^{Fc} = (C, F_C, \{(m_i, f_{mi})\}, D_C, D_L, h^{Dc}, h^{D_L}, par^{Dc}, par^{D_L}, Pred)$ *où :*

- *C désigne une constellation,*
- F_C *est le fait visualisé de la constellation et* $\{(m_i, fm_i)\}$ *l'ensemble des mesures visualisées du fait accompagnées de leurs fonctions d'agrégation,*
- D_C *et* D_L *sont deux dimensions liées au fait* F_C *représentées respectivement sur les colonnes et les lignes de la table,*
- $h^{Dc}{}_i$ *et* h^{D_L} *représentent les deux ensembles de hiérarchies des dimensions* D_C *et* D_L *selon lesquelles les données sont visualisées,*
- par^{Dc} *et* par^{D_L} *désignent les listes des paramètres visualisés en lignes et en colonnes,*
- *Pred est un ensemble de prédicats appliqués aux dimensions et au fait analysés afin de restreindre l'analyse en fonction des valeurs des paramètres et des mesures.*

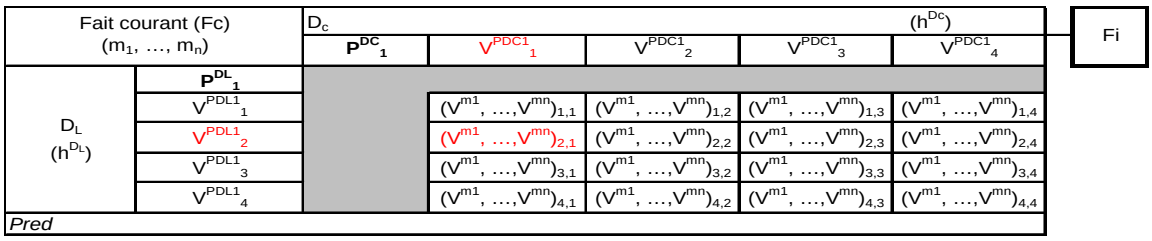


Figure III.4 *: Représentation graphique d'une table dimensionnelle*

2.2. Opérateurs dimensionnels intégrant les contraintes

Nous proposons dans cette section les différents opérateurs de notre langage d'interrogation dimensionnel qui tient compte de la sémantique des contraintes intra et inter-dimensions définies dans notre modèle dimensionnel. Nous distinguons les opérateurs de visualisation, de transformation de granularité tel que le forage et les opérateurs de transformation de la structure.

Une analyse dimensionnelle est une interrogation dynamique et interactive du schéma dimensionnel. L'opérateur *DISPLAY* ayant en entrée un schéma en constellation, permet de visualiser une première table dimensionnelle sur laquelle se basera la suite de l'analyse.

- ***Définition des opérateurs de visualisation (Display et HDisplay)***

L'opérateur de visualisation appliqué sur une constellation permet de visualiser une table dimensionnelle comportant un fait analysé en fonction de plusieurs dimensions. L'utilisateur indique les deux dimensions dont le contenu est visualisé en lignes et en colonnes, les autres dimensions se retrouvent en plan. Au niveau du fait, si le décideur n'indique pas les mesures à visualiser, toutes les mesures seront affichées.

Au niveau des dimensions, nous proposons aux décideurs deux types de visualisations :

a. une visualisation libre, sans indiquer les hiérarchies, réalisée par l'opérateur ***Display***. Le décideur indique les paramètres de l'analyse de chaque dimension. Ces paramètres peuvent appartenir à différentes hiérarchies. Dans la table dimensionnelle, nous affichons l'ensemble des hiérarchies qui passe par chaque paramètre.

Définition

L'opérateur ***Display*** *permet de visualiser une table dimensionnelle en fonction de deux paramètres sans fixer les hiérarchies. La syntaxe de l'opérateur est la suivante :*

$\mathbf{TD}^{Fc}$ = **Display** *(C,* F_C *, [{(*m_i*,* fm_i*)},]* D_C*,* D_L*,* p^{Dci}*,* p^{DLj} *[, Pred]) où :*

- *C désigne une constellation.*
- F_C *est le fait courant de la constellation*
- *{(*m_i*,* fm_i*)} l'ensemble des mesures visualisées accompagnées de leurs fonctions d'agrégation. Par défaut, nous affichons l'ensemble des mesures du fait associées à la fonction d'agrégation somme.*
- D_C *et* D_L *sont deux dimensions liées au fait* F_C *représentées respectivement sur les colonnes et les lignes de la table.*
- p^{Dci} *et* p^{DLj} *sont les deux paramètres des dimensions* D_C *et* D_L *fixés.*
- *Pred est un ensemble de prédicats qui sont appliqués sur les dimensions et les mesures de l'analyse.*

Exemple 2

Nous reprenons l'exemple du chapitre II décrivant les agences de location. Nous rappelons dans la Figure III.5 le schéma dimensionnel en constellation de cet exemple.

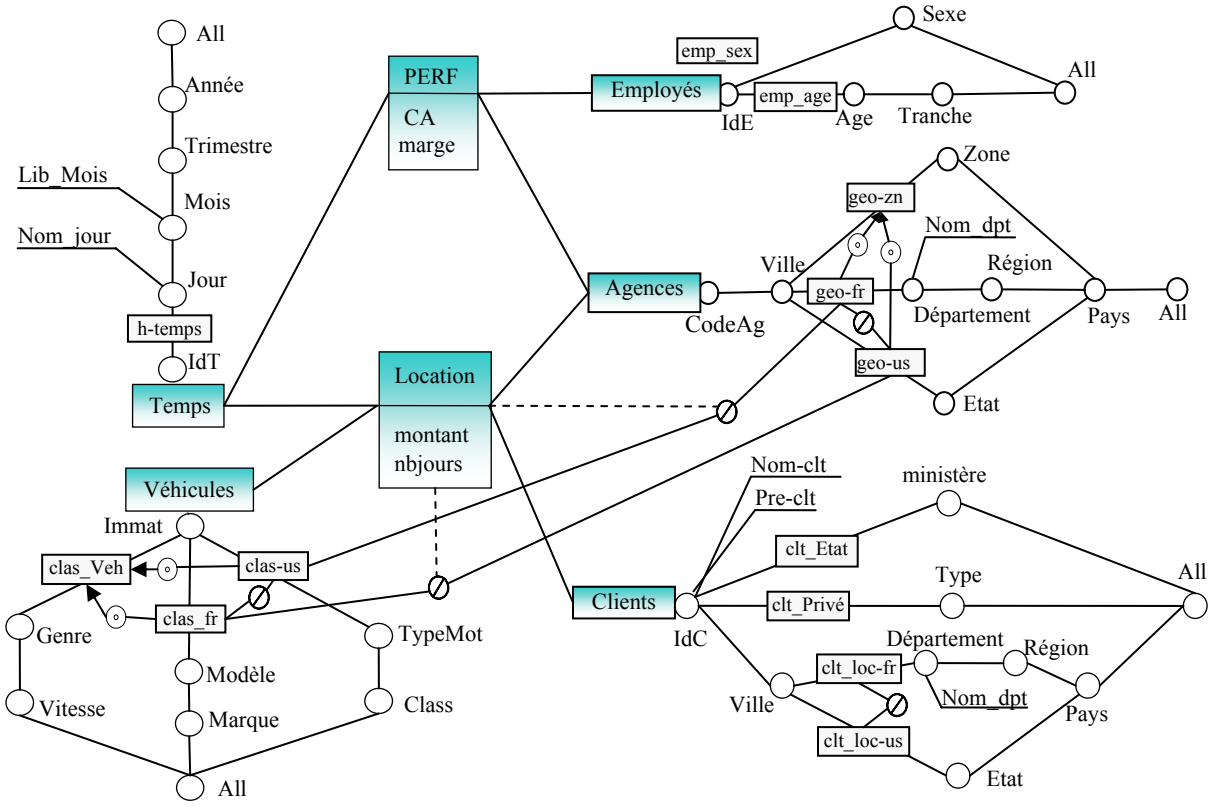


Figure III.5*: Représentation graphique d'une constellation.*

Le décideur souhaite visualiser les locations par *Pays* et par *Année* pour toutes les agences sans fixer les hiérarchies. Pour cela, il peut appliquer l'opérateur ***Display*** en visualisation libre. On exprime l'opérateur par :

TD[1] = **Display** (Location Véhicule, Location, Agences, Temps, Pays, Année)

Par abus de notation, on désigne les éléments du modèle (constellation, faits, dimensions…) par leur nom. Le résultat de l'opérateur est visualisé sous la forme d'une table dimensionnelle représentée dans la Figure III.6.

Location (*montant, nbJours*)		*Agences* Pays	*(géo_zn, géo_fr, géo_us)* France	Etats-Unis	PERF
Temps (*h_temps*)	Année				
	2002		(200, 20)	*(340, 25)*	
	2001		(150, 12)	*(650, 32)*	
	2000		(220, 25)	*(420, 28)*	
Véhicules.All='All' *Clients.All='All'*					

Figure III.6 *: Exemple de table dimensionnelle visualisée avec l'opérateur DISPLAY.*

Dans la Figure III.6, nous affichons toutes les hiérarchies qui passent par le paramètre *Pays* : "*geo_zn*", "*geo_fr*", "*geo_us*". En effet, les pays relatifs à ces trois hiérarchies sont visualisés dans la table.

b. une visualisation, basée sur les hiérarchies, réalisée par l'opérateur ***HDisplay***. Le décideur fixe les deux hiérarchies selon lesquelles les données seront visualisées.

Dans ce cas, les paramètres de l'analyse ne sont pas définis par le décideur. Par défaut, le paramètre de granularité maximale de chaque hiérarchie est affiché.

Définition

L'opérateur ***HDisplay*** *permet de visualiser une table dimensionnelle en fonction de deux hiérarchies. La syntaxe de l'opérateur est la suivante :*

TD^{Fc} =***HDisplay*** *(C, F_C , [{(m_i, fm_i)},] D_C, D_L , h^{Dci} , h^{DLj} [, Pred]) où :*

- *C désigne une constellation.*
- *F_C est le fait courant de la constellation*
- *{(m_i, fm_i)} l'ensemble des mesures visualisées accompagnées de leurs fonctions d'agrégation. Par défaut, nous affichons l'ensemble des mesures du fait associées à la fonction d'agrégation somme.*
- *D_C et D_L sont deux dimensions liées au fait F_C représentées respectivement sur les colonnes et les lignes de la table.*
- *h^{Dci} et h^{DLj} sont les hiérarchies visualisées des dimensions D_C et D_L.*
- *Pred est un ensemble de prédicats qui sont appliqués sur les dimensions et les mesures de l'analyse.*

Exemple 3

Supposons que le décideur souhaite analyser les locations en se basant sur les hiérarchies d'analyse. Il désire afficher les locations par pays et par année, en fonction des hiérarchies "*geo_fr*" et "*h_temps*". Pour cela, il peut appliquer l'opérateur *HDISPLAY*. On exprime l'opérateur par :

TD^2 = **HDisplay** (Location Véhicule, Location, Agences, Temps, geo_fr, h_temp)

Location (montant, nbJours)	*Agences*	*géo_fr*	PERF
	Pays	France	
Temps (h_temps)	Année		
	2002	(200, 20)	
	2001	(150, 12)	
	2000	(220, 25)	
Véhicules.All='All' *Clients.All='All'*			

Figure III.7 *: Visualisation de table dimensionnelle suivant les hiérarchies.*

Dans cet exemple, nous remarquons que les instances où le pays est égal à la valeur 'États-Unis' ne sont plus visualisées. C'est la condition d'appartenance à la hiérarchie "*geo_fr*", exprimée par notre modèle, qui permet de sélectionner les données à afficher.

- ***Contraintes et opérateurs de visualisation (HDisplay)***

Une première implication des contraintes inter-dimensions concerne l'opérateur de visualisation suivant des hiérarchies, ***HDisplay***. En effet, la visualisation des données dimensionnelles suivant deux hiérarchies en exclusion, renvoie des valeurs nulles au niveau des mesures qui peuvent être mal analysées par l'utilisateur. Dans ce cas, le système doit informer l'utilisateur de l'incompatibilité des deux hiérarchies. Par exemple, la visualisation des locations en fonction des dimensions *Agences* et *Véhicules*, suivant les hiérarchies "*geo_fr*" et "*clas_us*", est interdite par notre langage en respectant la contrainte d'exclusion entre ces deux hiérarchies.

Par contre, une contrainte d'inclusion entre les deux hiérarchies visualisées implique que le résultat du *HDisplay* ne doit afficher que les données compatibles avec la hiérarchie

incluse. Un message, affiché en bas de la table dimensionnelle, informe l'utilisateur de l'existence de données non visualisées au niveau de la hiérarchie incluante (cf. Figure III.8).

Exemple 4

Le décideur souhaite analyser les locations en fonction des agences, suivant la perspective zone ("*geo_zn*"), et en fonction des véhicules, suivant la classification française ("*clas_fr*"). L'opérateur exprimant ce besoin est le suivant ;

TD^3 = **HDisplay** (Location Véhicule, Location, Agences, Véhicules, geo_zn, clas_fr)

Location (montant, nbJours)	*Agences* / Pays	*geo_zn* / France	PERF
Véhicules (clas_fr) / Marque			
Peugeot		(290, 30)	
Coitröen		(260, 27)	
Temps.All='All' *Clients.All='All'*			
la hiérarchie clas_fr est incluse dans geo_zn; les données de geo_zn sont incomplètes			

***Figure III.8** : Exemple de table dimensionnelle avec deux hiérarchies en inclusion.*

Dans cet exemple, nous remarquons que seule la valeur 'France' pour le paramètre *Pays* est visualisée alors que la hiérarchie "*géo_zn*" de la dimension *Agences* décrit aussi les locations relatives aux 'Etats-Unis'. En effet, ces locations ne peuvent pas êtres analysées en fonction de la hiérarchie "*clas_fr*" relatives à la classification française des véhicules. Ce fait est exprimé par la contrainte d'inclusion de la hiérarchie "*clas_fr*" dans "*geo_zn*". Un message en bas de la table dimensionnelle informe le décideur de l'existence de cette contrainte.

Ce premier opérateur de visualisation permet de renvoyer une table dimensionnelle qui servira comme entrée aux autres opérateurs dimensionnels. Ces opérateurs permettent de manipuler d'une manière interactive et dynamique la table dimensionnelle obtenue.

Nous proposons deux familles d'opérateurs :

- les opérateurs de transformation de la granularité de l'analyse, tels que les opérateurs de forage et de calcul des totaux (cube),
- les opérateurs de transformation de la structure de la table dimensionnelle, tels que les opérateurs de rotation (Rotate), de permutation (Switch), d'emboîtement (Nest), etc.

Nous étudions ces deux familles d'opérateurs dans les sections suivantes.

2.2.1. Opérateurs de transformation de la granularité des données

Ces opérateurs réalisent le changement des paramètres d'analyse de la table dimensionnelle en passant par une granularité d'analyse plus ou moins fine.

2.2.1.1. Opérateurs de forage

Les opérateurs de forage permettent de visualiser les données dimensionnelles à des niveaux différents de détail. Ces opérateurs permettent de passer d'un paramètre de granularité donnée à un autre paramètre de granularité plus ou moins fine.

♦ **Contraintes et opérateurs de forage**

Soient *D*im une dimension et h_1 et h_2 deux hiérarchies pouvant être en exclusion, inclusion, partition ou totalité (cf. Figure III.9). Le forage à partir d'un paramètre commun à h_1 et h_2 (ex. P^D_1) vers un paramètre spécifique à h_1 (ex. P^D_2) (respectivement h_2, ex. P^D_3),

implique que seules les données de la hiérarchie h_1 (♣) (respectivement h_2 ♠) seront visualisées.

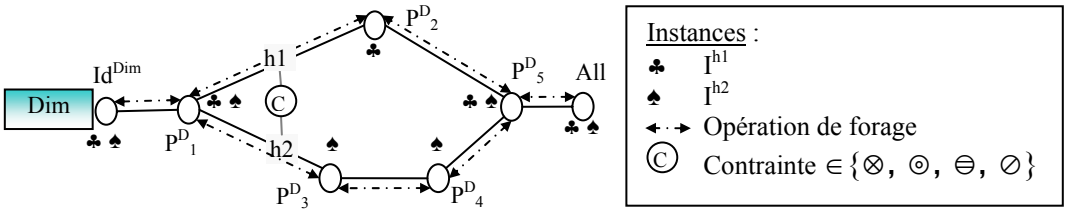


***Figure III.9** : Opération de forage sous une contrainte intra-dimension*

Par contre, le forage à partir d'un paramètre spécifique à h_1 (ex. P^D_2) (respectivement h_2 ex. P^D_3), vers un paramètre commun à h_1 et h_2 (ex. P^D_1) implique :

- soit le maintien de l'analyse sur le sous-ensemble des données de h_1 (♣) (respectivement h_2 ♠) initialement visualisées,
- soit l'ajout des données issues de la nouvelle hiérarchie h_2 (respectivement h_1) et donc réaliser une nouvelle analyse. L'agrégation des valeurs des mesures vers le paramètre commun doit prendre en compte l'union des instances des deux hiérarchies (♣ ♠).

♦ ***Définition du forage vers le bas (DrillDown)***

Définition

L'opérateur de forage vers le bas, ***DrillDown****, incrémente le nombre de paramètres en introduisant un niveau de détail plus fin appartenant à une hiérarchie visualisée d'une dimension. Le sens de forage part du niveau le plus général All vers le niveau le plus détaillé Id. La syntaxe de l'opérateur est la suivante :*

$\boldsymbol{TD}^S$ = **DrillDown** *(TD^E, D_i, P^S, [flag]) où :*

- *TD^E= (C, F^E, {(m_i^E, fm_i^E)}, D_C^E, D_L^E, h^{DLE}, h^{DcE}, par^{DLE}, par^{DcE}, $Pred^E$) est la table d'entrée.*
- *D_i est une dimension liée au fait visualisé F_C, avec P^{Ef} le paramètre le plus fin de D_i visualisé dans la table d'entrée.*
- *P^S est un paramètre de la dimension D_i de niveau plus fin que P^{Ef} tel que $P^{Ef} \in Param^{Di}(P^S)$.*
- *TD^S = (C, F^E, {(m_i^E, fm_i^E)}, D_C^E, D_L^E, h^{DLS}, h^{DcE}, $P^S \cup par^{DLE}$, par^{DcE}, $Pred^E$) est la table de résultat si la dimension de forage est en lignes.*
- *TD^S = (C, F^E, {(m_i^E, fm_i^E)}, D_C^E, D_L^E, h^{DLE}, h^{DcS}, par^{DLE}, $P^S \cup par^{DcE}$, $Pred^E$) est la table de résultat si la dimension de forage est en colonnes.*
- *flag est une variable booléenne qui indique si on maintient (flag=faux valeur par défaut) ou bien si on reprend une nouvelle analyse (flag=vrai) (cf. Figure III.10).*

Exemple 5

A partir de la table $\boldsymbol{TD}^1$, le décideur souhaite visualiser la table dimensionnelle regroupant les ventes par *Région*. Ce paramètre étant spécifique à la hiérarchie "*geo-fr*", le décideur ne peut que visualiser les données de cette hiérarchie. Le changement de hiérarchie, dans ce cas, est réalisé automatiquement par le système après avoir affiché un message informant le décideur de ce changement. L'opérateur de forage est exprimé comme suit :

TD^4= DrillDown (TD^1, Agences, Région, Faux)

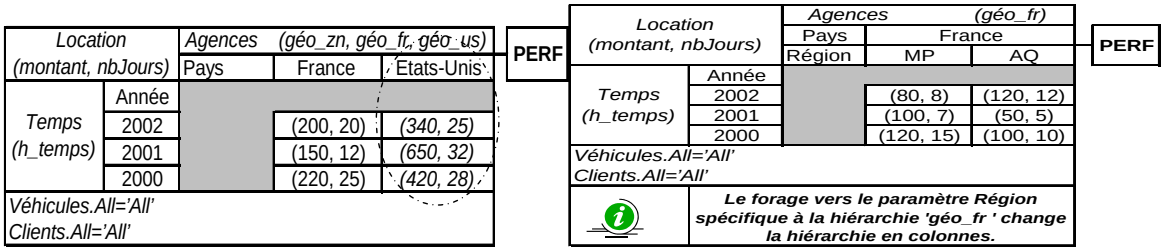


Figure III.10 *: Restriction de l'analyse ; forage vers un paramètre spécifique.*

Dans cet exemple, le paramètre optionnel *flag* n'intervient pas dans l'interrogation. C'est un cas de restriction d'analyse en passant d'un paramètre commun à deux hiérarchies (*Pays*) vers un paramètre spécifique (*Région*). L'opérateur de forage se base sur la condition d'appartenance aux hiérarchies pour définir l'ensemble valide des instances à afficher.

- ***Définition de forage vers le haut (RollUp)***

Définition

L'opérateur de forage vers le haut, ***RollUp****, diminue le nombre de paramètres et permet de passer à un niveau de détail moins fin dans une dimension. La syntaxe de l'opérateur est la suivante :*

$\mathbf{TD}^S$ = **RollUp** $(TD^E, D_i, P_S, [flag])$ *où :*

- TD^E *est la table d'entrée.*
- *Di est une dimension liée au fait visualisé* F_C*, avec* P^{Ef} *le paramètre le plus fin de* D_i *visualisé dans la table d'entrée.*
- P^S *est un paramètre de la dimension* D_i *de niveau moins fin que* P^{Ef} *avec* $P^S \in Param^{Di}(P^{Ef})$*. Nous notons* P^{inf} *l'ensemble des paramètres de* D_i *de niveau plus fin que* P^S *tel que* $P^{inf} = \{P^k \mid P^S \in Param^{Di}(P^k)\}$*.*
- $TD^S = (C, F^E, \{(m_i^E, fm_i^E)\}, D_C^E, D_L^E, h^{DLS}, h^{DcE}, P^S \cup par^{DLE} - P^{inf}, par^{DcE}, Pred^E)$ *est la table de résultat si la dimension de forage est en lignes.*
- $TD^S = (C, F^E, \{(m_i^E, fm_i^E)\}, D_C^E, D_L^E, h^{DLE}, h^{DcS}, par^{DLE}, P^S \cup par^{DcE} - P^{inf}, Pred^E)$ *est la table de résultat si la dimension de forage est en colonnes.*
- *flag est une variable booléenne qui indique si on maintient (flag=faux valeur par défaut) ou bien si on reprend une nouvelle analyse (flag=vrai).*

Exemple 6

Après le forage vers le paramètre *Région* (Figure III.10), le décideur souhaite revenir à la table dimensionnelle regroupant les ventes par *Pays*. Il dispose alors de deux choix : maintenir l'analyse aux données appartenant à la hiérarchie française "*geo-fr*", ou bien étendre l'analyse à toutes les agences à n'importe quel pays et à toutes les hiérarchies qui comportent ce paramètre. Supposons que le décideur souhaite étendre l'analyse et visualiser tous les *Pays*. L'opérateur de forage est exprimé comme suit :

$\mathbf{TD}^1$= **RollUp** (**TD**^{4}**,** Agences, Pays, **vrai**)

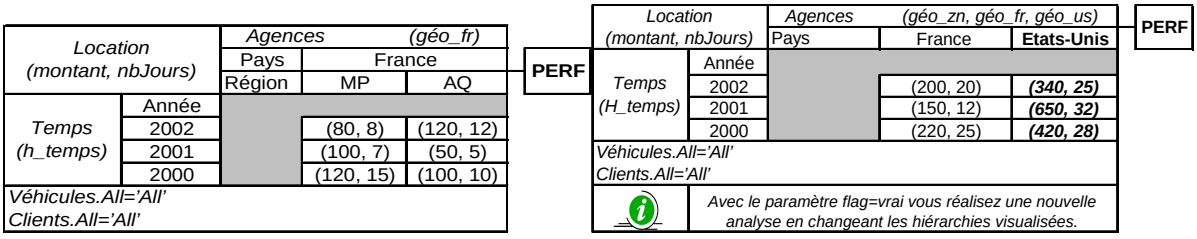


Figure III.11 *: Nouvelle analyse ; forage vers un paramètre commun.*

Dans cet exemple, nous proposons de réaliser une nouvelle analyse en appliquant l'opérateur *RollUp* étendu. Le système détecte que le nouveau paramètre *Pays* est commun à plusieurs hiérarchies et les affiche en informant l'utilisateur du changement de hiérarchies.

2.2.1.2. Opérateur de calcul des totaux

L'opérateur Cube appliqué sur une table dimensionnelle permet d'afficher les résultats d'agrégations des mesures du fait visualisés en fonction des différentes dimensions affichées en lignes et en colonnes.

- **Contraintes et opérateur de calcul des totaux (Cube)**

L'application de l'opérateur cube revient à afficher les totaux et les sous-totaux pour chaque dimension de la table dimensionnelle. Ainsi, les totaux en lignes sont calculés en agrégeant les valeurs en colonnes et inversement pour les totaux en colonnes (cf. Figure III.13). L'application de cet opérateur sur des hiérarchies en exclusion, inclusion ou partition implique :

- soit le maintien de l'analyse sur le sous-ensemble des données initialement visualisé,
- soit la visualisation des totaux de toutes les données de la dimension.

- **Définition de l'opérateur Cube**

Définition

L'opérateur ***Cube*** *appliqué sur une table dimensionnelle permet d'afficher les totaux et les sous totaux en lignes et en colonnes. La syntaxe de l'opérateur est la suivante :*

$\boldsymbol{TD}^S$ = ***Cube*** $(TD^E, [flag])$

- TD^E *est la table d'entrée.*
- $TD^S = (C, F^E, \{(m_i^E, f_{mi}^E)\}, D_C^E, D_L^E, h^{DcE}, h^{D_LE}, par^{\mathbf{DcE}} \cup \{All\ (H^{DcE})\}, par^{\mathbf{D_LE}} \cup \{All\ (D_L^E)\}, Pred^E)$ *est la table de résultat.*
- *flag désigne la variable booléenne qui indique si on maintient l'analyse ou bien si on reprend une nouvelle analyse (flag=vrai).*
- $All(D_I^E)$ *désigne le résultat de l'agrégation des mesures d'activité en considérant toutes les données de la dimension* D_I^E.

Exemple 7

En partant de la table dimensionnelle analysant les locations par *Région, Pays* et *Année* (TD^4), le décideur souhaite visualiser les totaux et les sous totaux correspondant aux montants des locations par *Région*, *Pays* et *Année*. L'opérateur ***CUBE***, sans tenir compte de l'extension, permet de visualiser ces données à l'aide de la syntaxe suivante :

$\mathbf{TD}^5$ = **Cube** (TD^4, Faux**)**

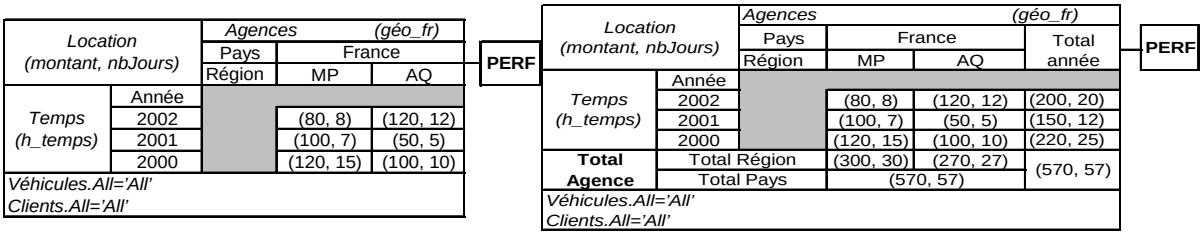


***Figure III.12** : Application de l'opérateur Cube.*

Le décideur souhaite visualiser tous les totaux des données relatives à la dimension *Agences* afin de pouvoir comparer le montant annuel des locations réalisées dans les agences françaises et le montant annuel des locations de toutes les agences. Ce besoin est obtenu en appliquant l'opérateur Cube étendu sur la table TD^4. Cet opérateur est exprimé à l'aide de l'expression suivante :

$$\mathbf{TD}^6 = \mathbf{Cube}\ (\mathrm{TD}^4, \textit{vrai})$$

Le résultat de cet opérateur étendu fait apparaître une nouvelle colonne comportant les totaux des locations par année pour toutes les agences (cf.Figure III.13).

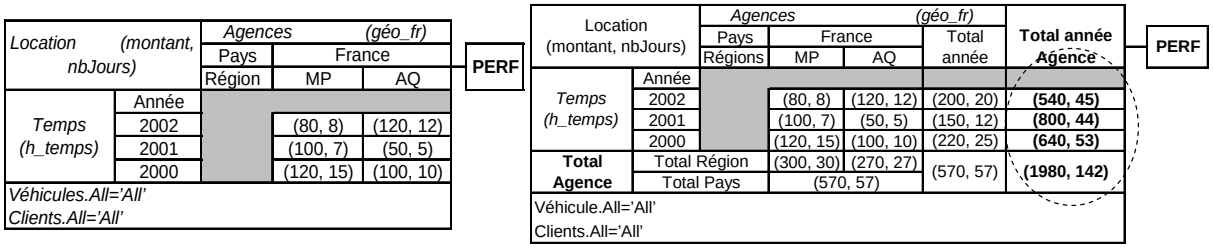


***Figure III.13** : Extension de l'opérateur Cube.*

2.2.2. Opérateurs de transformation de la structure des données

Les opérateurs de transformation permettent de changer la représentation du schéma dimensionnel. L'objectif de ces opérateurs est de manipuler le schéma afin de mieux appréhender les informations.

2.2.2.1. Opérateurs de rotation

L'opérateur de rotation, classique en analyse dimensionnelle, permet d'analyser les données selon différents axes et perspectives. Cet opérateur s'applique aux dimensions et aux hiérarchies ; il consiste à permuter deux dimensions ou deux hiérarchies afin de changer les paramètres utilisés pour visualiser les mesures d'activité. Une extension de cet opérateur a permis de l'appliquer sur des faits dans un modèle dimensionnel en constellation (Ravat et al, 2002).

- **Définition de la rotation des hiérarchies (*HRotate*)**

L'opérateur de rotation des hiérarchies, noté ***HRotate***, consiste à changer la hiérarchie utilisée pour visualiser les données selon une nouvelle perspective d'analyse dans une même dimension.

♦ Contraintes et opérateur de rotation des hiérarchies

Soient *D*im une dimension et h_1 et h_2 deux hiérarchies de *D*im pouvant être en exclusion, inclusion, partition ou totalité (voir Figure III.14). La rotation de h_1 vers h_2 implique :

- soit le maintien des données de l'analyse (♣), ce choix est possible dans le cas où h_1 et h_2 ne sont pas en exclusion ou en partition ($\clubsuit \cap \spadesuit \neq \varnothing$).
- soit le changement de l'analyse,
- Si h_1 est incluse dans h_2 ($\clubsuit \subseteq \spadesuit$) alors le changement correspond à une extension à toutes les données contenues dans la nouvelle hiérarchie.
- Si une contrainte de partition ou d'exclusion est définie entre les deux hiérarchies ($\clubsuit \cap \spadesuit = \varnothing$) alors c'est obligatoirement une nouvelle analyse.

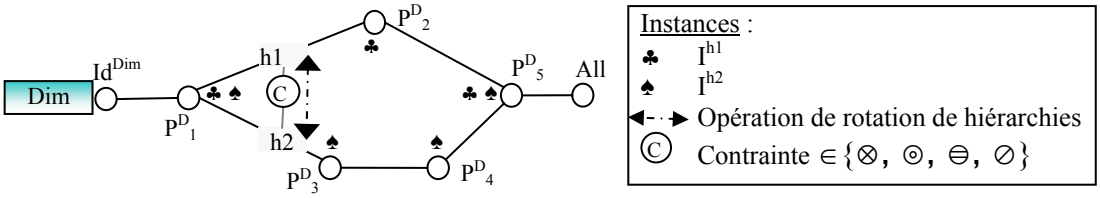


Figure III.14 *: Opération de rotation de hiérarchies sous une contrainte intra-dimension*

Définition

L'opérateur de rotation de hiérarchies, ***HRotate****, permet le changement de la perspective d'analyse en changeant la hiérarchie de visualisation en ligne ou en colonne.*

TD^S = ***HRotate*** *(*TD^E*,* D_m*,* h^{Dm}_j *[, flag]) où*

- TD^E *est la table d'entrée.*
- TD^S *= (C,* F^E*, {(*m_i^E*,* fm_i^E*)},* D_m*,,* DL^E*,* h^{Dm}_j*, {*h^{DLE}*}, {*p^{hDmj}*}, {*par^{DLE}*}), PredE) est la table de résultat. Nous notons que la hiérarchie en rotation peut appartenir à la dimension en lignes ou en colonnes.*
- D_m *est une dimension liée au fait visualisé* F^E*,*
- $h^{Dm}_j \in H^{Dm}$ *est une hiérarchie définie sur la dimension* D_m*,*
- p^{hDmj} *est le paramètre le moins détaillé de la hiérarchie* h^{Dmj}*.*
- *flag est un paramètre optionnel indiquant si l'opérateur est réalisé en maintenant l'analyse (flag=faux, valeur par défaut) ou en réalisant une nouvelle analyse (flag=vrai).*

Exemple 8

On considère la table dimensionnelle représentant l'analyse des locations par *Région* et par *Année (*TD^4*)*. Le décideur souhaite changer la hiérarchie "*geo_fr*" de la dimension *Agences* afin de visualiser les données selon la perspective offerte par la hiérarchie "*geo_zn*" en réalisant une nouvelle analyse. Cette opération peut être exprimée par l'expression suivante :

TD^7 = ***HRotate*** *(*TD^4*, Agences, geo_zn, vrai)*

Le résultat de l'opérateur est visualisé sous la forme d'une table dimensionnelle représentée dans la Figure III.15.

Location (montant, nbJours)	Agences (géo_fr)		
	Pays	France	
	Région	MP	AQ
Temps (h_temps) — Année			
2002		(80, 8)	(120, 12)
2001		(100, 7)	(50, 5)
2000		(120, 15)	(100, 10)
Véhicules.All='All' Clients.All='All'			

PERF

Location (montant, nbJours)	Agences (géo_zn)				
	Zone	Sud-Fr	Est-Fr	Sud-US	Est-US
Temps (h_temps) — Année					
2002		(150, 11)	(178, 22)	(142, 21)	(164, 20)
2001		(190, 15)	(110, 10)	(150, 15)	(190, 27)
2000		(160, 32)	(145, 16)	(145, 32)	(176, 33)
Véhicules.All='All' Clients.All='All'					

PERF

La rotation de la hiérarchie 'géo_fr' effectue une nouvelle analyse : de nouvelles données relatives à 'géo_zn' sont affichées.

***Figure III.15** : Résultat de l'opérateur de rotation de hiérarchies*

Cette rotation s'applique sur des hiérarchies pour lesquelles une contrainte d'inclusion est définie ("*geo_fr*" ⊙ "*geo_zn*"). L'application d'une opération de rotation avec l'option *flag* = *vrai* a permis d'étendre l'analyse des agences situées en France aux données des agences situées aux Etats-Unis (zones 'Sud-US' et 'Est-US').

♦ Définition de la rotation des dimensions (*DRotate*)

L'opérateur de rotation des dimensions, noté ***DRotate***, permet de modifier un axe d'analyse, c'est-à-dire de changer une dimension parmi celles utilisées pour visualiser les données d'un fait.

♦ Contraintes et opérateur de rotation des dimensions

Soient Dim_1 et Dim_2 deux dimensions et h_1 et h_2 deux hiérarchies appartenant respectivement à Dim_1 et Dim_2 telles que h_1 et h_2 pouvant être en exclusion, inclusion, partition ou totalité (voir Figure III.16). La rotation des dimensions de Dim_1 vers Dim_2 en passant de la hiérarchie h_1 vers h_2 implique :

– soit le maintien des données de l'analyse (◆♣), ce choix est possible dans le cas où h_1 et h_2 ne sont pas en exclusion ou en partition ($((◆♣) \cap (◆♠) \neq \varnothing)$;

– soit le changement de l'analyse,

- Si h_1 est incluse dans h_2 ($(♦♣) \subseteq (♦♠)$) alors le changement correspond à une extension à toutes les données associées à la nouvelle hiérarchie (♦♠).
- Si une contrainte de partition ou d'exclusion est définie entre les deux hiérarchies ($(♦♣) \cap (♦♠) = \varnothing$) alors c'est le changement complet de l'analyse. Les données correspondant à la première hiérarchie (♦♣) sont masquées et la table affiche les données relatives à la nouvelle hiérarchie (♦♠).

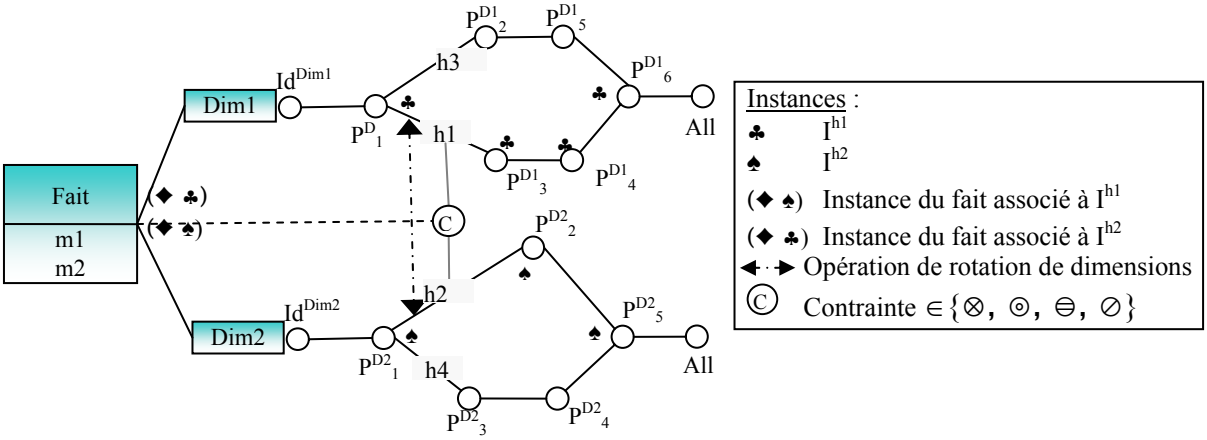


***Figure III.16** : Opération de rotation de dimensions sous une contrainte inter-dimensions*

De manière analogue à l'opérateur de rotation des hiérarchies, nous proposons un opérateur ***DRotate*** qui permet ou non le maintien de l'analyse lors de rotations mettant en jeu des données pour lesquelles des contraintes sémantiques sont définies.

Définition

L'opérateur de rotation de dimension, ***DRotate****, réalise le changement d'un axe d'analyse en remplaçant une dimension par une autre dans la table dimensionnelle. Le décideur peut optionnellement définir la hiérarchie visualisée de la nouvelle dimension. Par défaut, la dimension est positionnée sur le paramètre le plus fin (Id).*

TD ^{S} ***= DRotate*** *(*TD^E*,* D_i*,* D_j*, [*h^{Dj}*], [flag]) où*

- TD^E *est la table d'entrée.*
- D_i *et* D_j *respectivement l'ancienne et la nouvelle dimension visualisée.*
- h^{Dj} *est un paramètre optionnel permettant de fixer la hiérarchie de l'analyse.*
- $TD^S = (C, F^E, \{(m_i^E, fm_i^E)\}, D_C^E, D_j, h^{DcE}, \{h^{Dj}{}_n\}, par^{DcE}, p^{Dj}, Pred^E)$ *est la table résultat avec :*
- $\{h^{Dj}{}_n\}$ *est l'ensemble des hiérarchies visualisées sur la dimension* D_j *; cet ensemble est limité à un singleton* h^{Dj} *si le décideur fixe la hiérarchie de l'analyse, sinon cet ensemble correspond à toutes les hiérarchies de la dimension.*
- $p^{Dj} \in P^{Dj}$ *correspond à la granularité la moins fine de la hiérarchie si le décideur fixe la hiérarchie sinon il correspond au paramètre le plus fin de la dimension.*
- *flag est un paramètre optionnel indiquant si l'opération est réalisée en maintenant l'analyse (flag=faux, valeur par défaut) ou non (flag=vrai).*

Exemple 9

On considère la table dimensionnelle TD^4. Un décideur souhaite poursuivre l'analyse en effectuant une visualisation des ventes non plus selon la dimension *Agences* mais en utilisant la dimension *Véhicules*. Cette opération de rotation est exprimée par l'expression algébrique suivante :

TD^8 = **DRotate** (TD^4, Agences, Véhicules, Clas_Veh, **faux**)

Cette rotation s'applique sur des dimensions dont les hiérarchies sont en inclusion (geo_*fr* ⊙ *Clas_Veh*) puisque toutes les instances du fait *Location* réalisées dans des agences françaises (organisées selon la hiérarchie "geo_*fr*"*)* sont liées à une instance de la hiérarchie *Clas_Veh* (de la dimension *Véhicules*).

Le décideur maintient l'analyse sur les instances concernant les agences françaises en utilisant l'option *flag = faux*. Ainsi, les données visualisées dans la Figure III.17 concernent les locations vendues par les agences françaises.

Location (montant, nbJours)		*Agences* (géo_fr)		
		Pays	France	
		Région	MP	AQ
Temps (h_temps)	Année			
	2002		(80, 8)	(120, 12)
	2001		(100, 7)	(50, 5)
	2000		(120, 15)	(100, 10)
Véhicules.All='All' *Clients.All='All'*				

PERF

Location (montant, nbJours)		*Véhicules* (Clas_Veh)		
		Vitesse	Gvitesse	Mvitesse
Temps (h_temps)	Année			
	2002		(120, 10)	(520, 45)
	2001		(60, 12)	(940, 123)
	2000		(20, 5)	(280, 110)
Agences.All='All' and Agences.Pays = 'France' *Clients.All='All'*				

PERF

Avec le paramètre flag=faux vous maintenez l'analyse aux données de geo_fr.

Figure III.17 *: Résultat de l'opérateur de rotation de dimensions.*

Notre modèle en constellation permet de définir plusieurs faits dans le même schéma. Par conséquent, nous proposons l'opération de rotation de sujets d'analyse (faits) de manière à pouvoir permuter les faits de notre constellation.

♦ ***Définition de l'opérateur de rotation des faits (FRotate)***

Définition

L'opérateur de rotation des faits, noté FRotate, permet de modifier le sujet d'analyse tout en gardant les différentes perspectives, c'est-à-dire de changer le fait et de garder les dimensions devant être communes aux deux faits permutés. La syntaxe de l'opérateur est la suivante :

$TD^S = FRotate\ (TD^E, F^S)$ *où :*

– F^S *est un fait de la constellation C.*

– TD^E *et* TD^S *sont des tables dimensionnelles d'entrée et de résultat.*

– $TD^S = (C, F^S, \{(m_i, fm_i)\}, D_C^E, D_L^E, h^{DcE}, h^{DLE}, par^{DcE}, par^{DLE}, Pred)$.

♦ ***Contraintes et opérateur de rotation des faits***

Les contraintes définies au niveau des hiérarchies inter et intra dimensions n'ont pas de répercussion sur la rotation des faits. En effet, la rotation des faits ne change pas les axes de l'analyse, ni les perspectives sur lesquelles portent les contraintes.

2.2.2.2. Opérateurs de restructuration des paramètres

Dans cette section, nous regroupons les opérateurs de changement de structure de la *TD*. Ces opérateurs réorganisent la position des paramètres des dimensions ou des valeurs de ces paramètres sans changer les mesures de l'analyse. Ils visent à mettre en avant un paramètre de l'analyse.

Pour définir ces opérateurs, qui peuvent intervenir au niveau de l'ordonnancement des valeurs des paramètres, nous définissons le concept suivant :

♦ ***Concept de précédence de valeur***

L'ordre des valeurs des paramètres dans la TD^{Fc} est prédéfini. Nous définissons le concept de ***précédence de valeur*** pour décrire cet ordre.

On introduit une relation d'ordonnancement entre les valeurs des attributs appartenant à l'ensemble des paramètres visualisés $param^{Dc}$ ou $param^{D_L}$ d'une table. Cette relation est définie par un prédicat de précédence $\boldsymbol{Prec_{Pi}(v^1_{Pi}, v^2_{Pi})}$ qui est vérifié si v^1_{Pi} précède v^2_{Pi} dans la table dimensionnelle avec v^1_{Pi} et v^2_{Pi} deux valeurs de l'attribut P_i. On note par exemple :

– $Prec_{Mois}$ (Janvier, Février)

– $Prec_{Année}$ (2000, 2001)

– $Prec_{Ville}$ (Toulouse, Montauban)

♦ ***Définition de la permutation (Switch)***

Cet opérateur permet de réorganiser une table dimensionnelle et de mettre en avant certaines valeurs de paramètres en permutant leurs positions dans la table.

Définition

L'opérateur ***Switch*** *permet de permuter les lignes et/ou les colonnes d'une table dimensionnelle en inversant les positions des valeurs d'un paramètre de la dimension. Il se définit par l'expression suivante :*

TD^S = Switch *(TD^E, P_i, v^1_{Pi} , v^2_{Pi})* ***où***

– *TD^E et TD^S sont des tables dimensionnelles d'entrée et de résultat tel que :*

dans TD^E , $P_i \in par^{DcE}$ ou $P_i \in par^{D_LE}$ et $Prec_{Pi}$ (v^1_{Pi} , v^2_{Pi}),

dans TD^S , $P_i \in par^{DcS}$ ou $P_i \in par^{D_LS}$ et $Prec_{Pi}$ (v^2_{Pi} , v^1_{Pi}).

– *v^1_{Pi} et $v^2_{Pi} \in DOM(P_i)$.*

Exemple 10

On considère la table dimensionnelle représentant l'analyse des locations par *Région* et par *Année*. Le décideur souhaite permuter les positions des valeurs 2000 et 2001 du paramètre *Année* afin de comparer les locations réalisées pendant les années 2002 et 2000. Cette opération peut être exprimée par l'expression suivante :

TD^9 = ***Switch*** (TD^4, Année, "2000","2001")

Le résultat de l'opérateur est visualisé sous la forme d'une table dimensionnelle représentée dans la Figure III.18.

Location (montant, nbJours)		Agences (géo_fr)		
		Pays	France	
		Région	MP	AQ
Temps (h_temps)	Année			
	2002		(80, 8)	(120, 12)
	2001		(100, 7)	(50, 5)
	2000		(70, 8)	(100, 10)
Véhcules.All='All' Clients.All='All'				

PERF

Location (montant, nbJours)		Agences (géo_fr)		
		Pays	France	
		Région	MP	AQ
Temps (h_temps)	Année			
	2002		(80, 8)	(120, 12)
	2000		(70, 8)	(100, 10)
	2001		(100, 7)	(50, 5)
Véhicules.All='All' Clienst.All='All'				

PERF

Figure III.18 *: Résultat de l'opérateur de permutation.*

Notons qu'il n'y a pas de répercussions des contraintes sur cet opérateur puisqu'il ne manipule que les valeurs d'un seul paramètre.

♦ ***Définition de l'emboîtement (Nest)***

Cet opérateur permet de fournir une représentation bidimensionnelle du *cube* dimensionnel quel que soit le nombre de dimensions.

Définition

L'opérateur ***Nest*** *permet d'imbriquer les valeurs d'un paramètre avec un autre. Les deux paramètres peuvent appartenir à des hiérarchies et même à des dimensions différentes. La dimension d'emboîtement représente le premier niveau tandis que la dimension emboîtée représente un niveau secondaire.*

TD^S = Nest *(TD^E, P_i, P_j) où*

– *$P_j \in \cup_{n=1,\ m} \{P^{Dn}\}$ avec m le nombre de dimension dans C et P^{Dn} les paramètres de la dimension D_n.*

– *TD^E et TD^S sont des tables dimensionnelles d'entrée et de résultat tel que :*

$TD^S = (C, F_c^E, \{(m_i, fm_i)\}, D_C^E, D_L^E, h^{DcE}, h^{D_LE}, par^{DcE} \cup P_j, par^{D_LE}, Pred^E)$. P_j est imbriqué au dernier niveau de la liste des paramètres en lignes ou en colonnes de la dimension.

♦ ***Contraintes et opérateur d'emboîtement***

Selon la combinaison de paramètres à emboîter plusieurs cas d'étude sont possibles ;

– Le cas le plus simple consiste à emboîter deux paramètres de la même hiérarchie. Par exemple, nous pouvons emboîter le paramètre *Année* dans le paramètre *Trimestre* de façon à mettre en relief les données trimestrielles. Ceci permet de faciliter la comparaison des résultats d'un trimestre donné pour les différentes années.

Exemple 11

A partir d'une table dimensionnelle qui décrit les locations par année et par trimestre (TD^{10}), nous souhaitons visualiser les données trimestrielles par année. L'opérateur d'emboîtement permet de répondre à ce besoin en inversant les positions des paramètres *Année* et *Trimestre*. Cet opérateur peut être exprimé par l'expression algébrique suivante :

$\mathbf{TD}^{11} = \mathbf{Nest}$ (TD^{10}, Trimestre, Année)

Location (montant, nbJours)			*Agences (geo_fr)*			PERF
			Pays	France		
			Région	MP	AQ	
Temps (h_temps)	Année	Trimestre				
	2002	T1		(25, 3)	(30, 3)	
		T2		(20, 2)	(25, 3)	
		T3		(15, 1)	(33, 4)	
		T4		(20, 2)	(32, 2)	
	2001	T1		(23, 2)	(13, 1)	
		T2		(26, 1)	(16, 2)	
		T3		(31, 3)	(11, 1)	
		T4		(20, 1)	(10, 1)	
Véhicules.All='All' *Clients.All='All'*						

Location (montant, nbJours)			*Agences (geo_fr)*			PERF
			Pays	France		
			Région	MP	AQ	
Temps (h_temps)	Trimestre	Année				
	T1	2002		(25, 3)	(30, 3)	
		2001		(23, 2)	(13, 1)	
	T2	2002		(20, 2)	(25, 3)	
		2001		(26, 1)	(16, 2)	
	T3	2002		(15, 1)	(33, 4)	
		2001		(31, 3)	(11, 1)	
	T4	2002		(20, 2)	(32, 2)	
		2001		(20, 1)	(10, 1)	
Véhicules.All='All' *Clients.All='All'*						

Figure III.19 *: Emboîtement de deux paramètres de la même hiérarchie*

L'emboîtement de deux paramètres de la même hiérarchie ne présente pas de répercussions des contraintes.

– Le deuxième cas concerne l'emboîtement de deux paramètres de même dimension mais de hiérarchies distinctes. En considérant les différentes contraintes entre les hiérarchies, plusieurs possibilités de combinaisons sont offertes :

- l'emboîtement de deux paramètres de hiérarchies en exclusion ou en partition est impossible. Par exemple, l'analyse des locations en fonction des régions et des états américains n'a pas de sens, car les locations concernent l'un ou l'autre des deux paramètres et pas les deux à la fois. Ce fait est exprimé par la contrainte de partition entre les hiérarchies "geo_fr" et "geo_us" ;
- l'emboîtement de deux paramètres de hiérarchies distinctes entre lesquelles nous avons défini une contrainte de simultanéité revient à un emboîtement de deux paramètres de la même hiérarchie ;
- l'emboîtement de deux paramètres de hiérarchies en inclusion dépend du sens de l'inclusion entre ces deux hiérarchies.
    - ♦ Le passage de la hiérarchie incluse vers la hiérarchie incluante ne présente pas de répercussions. Par exemple, le passage de l'analyse des locations par région suivant la hiérarchie "*geo_fr*" vers l'analyse par *Région* et par *Zone* de la hiérarchie "*geo_zn*", revient à emboîter les zones dans les différentes régions. Cet emboîtement ne risque pas de causer des conflits puisque toutes les données de la

hiérarchie "*geo_fr*" peuvent être analysées suivant la hiérarchie "*geo_zn*" (inclusion de "*geo_fr*" dans "*geo_zn*").

- ♦ Par contre, le passage de la hiérarchie incluante vers la hiérarchie incluse nécessite de restreindre l'analyse aux données de la hiérarchie incluse.

Exemple 12

Considérons l'exemple d'une table dimensionnelle analysant les locations par pays selon la hiérarchie "*geo_zn*" (TD^{12}). Nous souhaitons combiner le paramètre *Région* de la hiérarchie *"geo_fr"* dans le paramètre *Pays*. Ceci est réalisé à l'aide de l'expression suivante :

$\mathbf{TD}^{13} = \mathbf{Nest}$ (TD^{12}, Pays, Région)

Location (montant, nbJours)		*Agences* (géo_zn)		
		Pays	France	Etats-unis
Temps (h_temps)	Année			
	2002		(200, 20)	(340, 25)
	2001		(150, 12)	(650, 32)
	2000		(120, 15)	(100, 10)
Véhicules.All='All' *Clients.All='All'*				

PERF

Location (montant, nbJours)		*Agences* (géo_zn)		
		Pays	France	
Temps (h_temps)	Année	Région	MP	AQ
	2002		(80, 8)	(120, 12)
	2001		(100, 7)	(50, 5)
	2000		(70, 8)	(50, 7)
Véhicules.All='All' *Clients.All='All'*				

PERF

Figure III.20 *: Emboîtement de deux paramètres de hiérarchies différentes*

Nous remarquons que l'emboîtement du paramètre *Régions* dans le paramètre *Pays* a nécessité la restriction de l'analyse aux régions françaises. Les données des États-unis ne sont plus visualisées puisqu'elles ne sont pas compatibles avec le nouveau paramètre.

– Le troisième cas concerne l'emboîtement de deux paramètres de dimensions distinctes permettant d'imbriquer plusieurs axes d'analyse dans la table dimensionnelle (Marcel, 1998). En considérant les différentes contraintes inter-dimensions, plusieurs possibilités de combinaisons sont offertes :

- l'emboîtement de deux paramètres appartenant à des hiérarchies sur lesquelles nous avons défini une contrainte de simultanéité, n'implique pas de répercussion sur l'analyse.

Exemple 13

A partir de la table dimensionnelle TD^4, nous souhaitons analyser les locations par *Région* et par *Marque* de véhicule. Ces deux paramètres appartiennent respectivement aux hiérarchies "*geo_fr*" de la dimension *Agences* et "*clas_fr*" de la dimension *Véhicules* entre lesquelles nous avons défini une contrainte de simultanéité. L'emboîtement des deux paramètres est réalisé par l'expression suivante :

$TD^{14} = \mathbf{\textit{Nest}}$ (TD^4, *Région, Marque*)

Location (montant, nbJours)		*Agences* (géo_fr)		
		Pays	France	
Temps (h_temps)	Année	Région	MP	AQ
	2002		(80, 8)	(120, 12)
	2001		(100, 7)	(50, 5)
	2000		(70, 8)	(100, 10)
Véhicules.All='All' *Clienst.All='All'*				

PERF

Location (montant, nbJours)		*Agences* (géo_fr)				
		Pays	France			
		Région	MP		AQ	
Temps (h_temps)	Année	Marque	Peugeot	Renault	Peugeot	Renault
	2002		(45, 5)	(35, 3)	(70, 7)	(50, 5)
	2001		(60, 4)	(40, 3)	(35, 3)	(15, 2)
	2000		(44, 5)	(26, 3)	(60, 7)	(40, 3)
Clients.All='All'						

PERF

Figure III.21 *: Emboîtement de deux paramètres de dimensions différentes*

La Figure III.21 présente un exemple d'emboîtement de paramètres appartenant à des dimensions différentes. Cette opération permet de représenter plusieurs dimensions (plus que deux) dans une table dimensionnelle.

- l'emboîtement de deux paramètres appartenant à des hiérarchies entre lesquelles nous avons défini une contrainte d'exclusion ou de partition est interdit. Par exemple, l'analyse des locations par Région, suivant la hiérarchie "geo_fr", et par Class, suivant la nomenclature américaine des véhicules, est interdite par le système car elle combine des données incompatibles ;
- l'emboîtement de deux paramètres de dimensions distinctes et de hiérarchies en inclusion dépend du sens de l'inclusion ;
    - ♦ Le passage de la hiérarchie incluse vers la hiérarchie incluante ne présente pas de répercussion.

Exemple 14

Le passage de l'analyse des locations par *marque* suivant la hiérarchie "*clas_fr*" de la dimension *Véhicules* vers l'analyse par *Marque* et par *Zone* de la hiérarchie "*geo_zn*", revient à emboîter les zones dans les différentes marques. Les montants des locations réalisés pour une marque de véhicule seront donc divisés entre les différentes zones géographiques les concernant. La hiérarchie d'inclusion de "*clas_fr*" dans "*geo_zn*" implique que toutes les données de la première hiérarchie peuvent être analysées suivant la deuxième et donc que l'opération d'emboîtement est valide pour toutes les données.

- ♦ Par contre, le passage de la hiérarchie incluante vers la hiérarchie incluse nécessite de restreindre l'analyse aux données de la hiérarchie incluse.

Exemple 15

Considérons l'exemple d'une table dimensionnelle analysant les locations par *pays* selon la hiérarchie "*geo_zn*" (*TD*[12]). Nous souhaitons visualiser les locations pour chaque marque de véhicule dans chaque pays. Ceci est réalisé à l'aide de l'expression suivante :

TD^{15} = **Nest** (TD^{12}, Pays, Marque)

Location	*Agences*		*(géo_zn)*
(montant, nbJours)	Pays	France	Etats-unis
Année			
Temps 2002		(200, 20)	(340, 25)
(h_temps) 2001		(150, 12)	(650, 32)
2000		(120, 15)	(100, 10)
Véhicules.All='All'			
Clients.All='All'			

PERF

Location	*Agences*	*(géo_zn)*	
(montant, nbJours)	Pays	France	
	Marque	Peugeot	Renault
Année			
Temps 2002		(115, 12)	(85, 8)
(h_temps) 2001		(95, 7)	(55, 5)
2000		(130, 15)	(90, 10)
Clients.All='All'			

PERF

***Figure III.22** : Emboîtement de deux paramètres sous contrainte d'inclusion inter-dimension.*

Nous remarquons que l'emboîtement du paramètre *Marque* dans le paramètre *Pays* a nécessité la restriction de l'analyse aux données analysées suivant la classification française des véhicules "*clas_fr*". Les données des 'États-unis' ne sont plus visualisées puisqu'elles ne sont pas compatibles avec le nouveau paramètre *Marque*.

2.3. Synthèse de l'impact des contraintes sur les opérateurs

Le tableau suivant présente l'impact des différentes contraintes sur les opérateurs dimensionnels.

	RollUp/ DrillDown	**Cube**	**HRotate**	**DRotate**	**Nest**
Intra-dimension					
Exclusion	*faux*	*Faux*	*faux*	*Vrai/ faux*	***Interdit***
Inclusion	*vrai/faux*	*Vrai/faux*	*vrai/ faux*	*vrai/ faux*	√
Simultanéité	*vrai/faux*	*vrai/faux*	*vrai/ faux*	*vrai/ faux*	
Totalité	*vrai/faux*	*vrai/faux*	*vrai/ faux*	*vrai/ faux*	√
Partition	*faux*	*faux*	*faux*	*vrai/ faux*	***Interdit***
Inter-dimensions					
Exclusion	*faux*	*faux*	*faux*	*faux*	***Interdit***
Inclusion	*vrai/faux*	*vrai/faux*	*Vrai/ faux*	*vrai/ faux*	√
Simultanéité	*vrai/faux*	*vrai/faux*	*vrai/ faux*	*vrai/ faux*	
Totalité	*vrai/faux*	*vrai/faux*	*vrai/ faux*	*vrai/ faux*	√
Partition	*faux*	*faux*	*faux*	*faux*	***Interdit***

Tableau III.1 *: Contraintes et opérateurs dimensionnels*

Nous avons proposé un ensemble d'opérateurs dimensionnels qui intègrent l'expression des contraintes sémantiques. Notamment, lors des manipulations des données dimensionnelles sous contraintes, nous proposons au décideur de choisir entre le maintien de l'ensemble des données analysées (*flag = faux*) ou la réalisation d'une nouvelle analyse (*flag = vrai*). Le Tableau III.1 présente les possibilités exprimées par *flag* (*vrai, faux*) pour chaque opérateur appliqué sur des données dimensionnelles reliées par un type donné de contraintes. Pour l'opérateur Nest, nous présentons l'impact des contraintes sur l'application de cet opérateur : (***Interdit***) pour indiquer que l'opérateur ne peut pas être appliqué dans ce cas et (√) pour indiquer que les contraintes permettent de calculer correctement l'ensemble des données.

3. Contraintes et vues matérialisées

L'impact des contraintes exprimées dans notre modèle dimensionnel, ne se limite pas au niveau de la définition des opérateurs dimensionnels mais s'étend à l'optimisation de ces opérateurs. Dans cette section, nous proposons d'intégrer la sémantique des contraintes dans le processus de sélection des vues matérialisées visant à diminuer le temps de réponse lors de l'interrogation des données dimensionnelles.

Nous définissons, dans un premier temps, les concepts de vues matérialisées et de treillis dimensionnel ainsi que la problématique de sélection des vues sous contraintes (section 3.1). Puis, nous construisons le treillis dimensionnel basé sur la structure hiérarchique des dimensions. Nous décrivons ensuite les étapes d'intégration des contraintes sémantiques dans le processus de construction du treillis dimensionnel représentant l'ensemble des vues à matérialiser (section 3.2).

3.1. Préliminaires

La structure hiérarchique des dimensions dans un modèle dimensionnel permet d'analyser les données en fonction de différentes combinaisons de paramètres. (Gray *et al.*, 1996). Ces combinaisons représentent les différents paramètres d'agrégation des données dimensionnelles détaillées. L'agrégation de ces données présente un coût très important qui diminue les performances du système OLAP. Ainsi, elles sont stockées sous forme de vues, appelées vues matérialisées, afin d'améliorer le temps d'interrogation du système (Gupta et al*,* 1999).

3.1.1. Concept de vue matérialisée

Une ***vue*** est une relation dérivée (virtuelle) construite à partir d'autres relations de la base de données. Une vue ainsi définie est calculée à chaque fois qu'elle est appelée. ***Une vue matérialisée*** est une vue dont les données sont stockées dans la base.

Le choix de ces vues présente un problème ; une vue matérialisée permet de diminuer le temps d'interrogation mais elle nécessite un coût de rafraîchissement et de stockage.

Dans nos travaux, l'objectif d'une vue matérialisée est d'effectuer un calcul de pré agrégats afin d'obtenir de meilleures performances en matière de temps de réponse. Une gestion dynamique de ces vues permet d'ajouter ou d'enlever des vues matérialisées ; leur stockage est temporaire en fonction de l'évolution de l'analyse.

3.1.2. Contraintes et Problème de Sélection des Vues matérialisées (PSV)

Le problème fondamental de la sélection des vues est de trouver l'ensemble des vues à matérialiser qui offre un meilleur compromis entre les temps de réponse aux requêtes des décideurs et le coût de maintenance des vues matérialisées.

Le but de notre étude est de comparer le résultat réalisé en sélectionnant les vues à matérialiser sans considération des contraintes sémantiques avec celui obtenu en intégrant ces contraintes. En effet, nous constatons que les vues qui ne satisfont pas les contraintes sémantiques du modèle dimensionnel n'apportent pas de gain de temps d'interrogation. Prenons l'exemple de la vue V combinant les ventes par départements français et états américains. En respectant la contrainte d'exclusion entre la hiérarchie de la géographie française et celle de la géographie américaine, nous obtiendrons une vue V vide. Supposons que cette vue contient des données, alors la BDM est incohérente et la contrainte d'exclusion n'est pas satisfaite. L'utilisateur peut demander de telles requêtes à la BDM et c'est le système qui doit détecter les incohérences. En outre, l'élimination de ces vues réduit considérablement la taille du treillis dimensionnel sur lequel se base les techniques d'optimisation de la sélection des vues matérialisées (Baralis et al, 1997) (Gupta et al, 1999) (Baril et al, 2003) (Paraboschi et al, 2003).

Nous proposons d'étudier le problème de sélection des vues en se basant sur le concept de treillis dimensionnel (Bellatreche, 2000).

3.1.3. Concept de treillis dimensionnel

Dans un modèle dimensionnel, les requêtes des décideurs se basent sur l'agrégation des mesures d'activité en fonction des différentes combinaisons des paramètres d'analyses. Ces combinaisons de paramètres forment un ensemble de vues dimensionnelles. Ces vues sont représentées par un ***treillis dimensionnel*** (Harinarayan et al, 1996). Chaque nœud du treillis représente une vue qui agrège les mesures de l'analyse en fonction d'une combinaison de paramètres (les paramètres du nœud). Chaque lien pointe du nœud i vers le nœud j si la vue représentée par j peut être agrégée à partir de la vue représentée par i. Par exemple, à partir du nœud (la vue) regroupant les locations par agence et par véhicule, représenté par la combinaison des identifiants respectifs des agences (*CodAg*) et des véhicules (*Immat*), nous pouvons calculer le nœud (la vue) qui regroupe les locations par agence, représenté par l'identifiant de l'agence (*CodAg***)** (nœuds grisés de la Figure III.23).

- Le nœud minimum correspond à la vue représentant les données de plus bas niveau.
- Le nœud maximum (*All*) correspond à la vue agrégée en fonction de tous les paramètres.

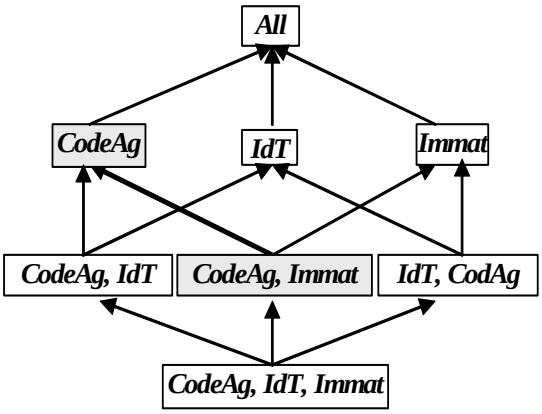


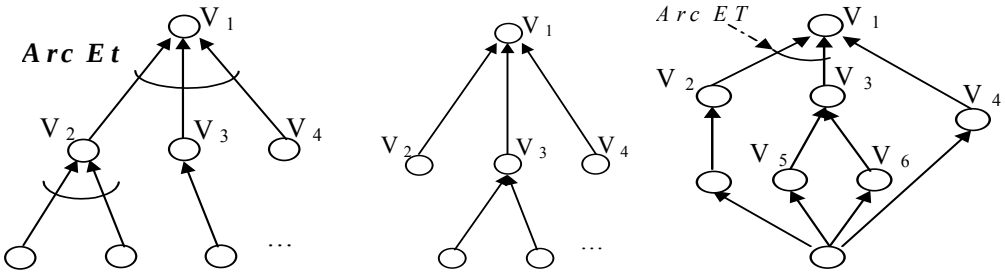
Figure III.23 *: Treillis du fait Location (Harinarayan et al, 1996)*

Trois types de treillis peuvent être construits selon la sémantique des relations entre les vues (Gupta et al, 1999) :

Un ***treillis*** de type ***ET***, défini sur l'ensemble des vues V, est un graphe acyclique dont les nœuds sont les vues de V. Un lien part de l'ensemble des nœuds v_1, v_2, …v_n, vers le nœud v_i si v_i peut être calculé à partir de l'ensemble v_1, v_2, …v_n et cette dépendance est présentée par un demi-cercle à travers les liens (v_i, v_1), (v_i, v_2) (v_i, v_n), appelé ***Arc ET***.

Un ***treillis*** de type ***OU*** est un graphe acyclique dont les nœuds sont les vues de V. Un Lien pointe du nœud v_j vers le nœud v_i si v_j peut être calculé à partir de v_i.

Un ***treillis*** de type ***ET-OU***, est un graphe acyclique dont les nœuds sont les vues de V. Chaque nœud peut avoir un ou plusieurs ***Arc ET*** et ***OU*** qui le relient aux autres nœuds à partir desquels il peut être calculé.



(a) Graphe ET (b) Graphe OU (c) Graphe ET-OU

Figure III.24 *: Graphes ET, OU et ET-OU*

Dans notre proposition, nous utilisons les treillis de type *ET-OU* afin d'exprimer la sémantique des contraintes d'exclusion, de partition et de totalité (cf. § 3.2). Nous présentons dans la Figure III.24 (c) un exemple de ce type de treillis combinant les graphes de type ET (a) et de type OU (b). Dans cet exemple, nous avons défini un arc ET sur les liens sortant des vues V_2 et V_3 vers V_1 exprimant le fait que V_1 est calculée à partir de l'intégration de V_2 et V_3. Un exemple de lien OU est représenté par les deux liens reliant V_5 et V_6 à V_3 indiquant que la vue V_3 peut être calculée à partir de l'une ou l'autre des deux autres vues.

La construction de ce treillis est une étape préliminaire qui permet par la suite de sélectionner l'ensemble des vues à matérialiser (Harinarayan et al, 1996) (Paraboschi et al, 2003). Dans notre exemple, la combinaison des dimensions *Agences* et *Véhicules* permet de construire un treillis qui combine deux à deux les hiérarchies des deux dimensions. Cette

combinaison pourra donner lieu à la définition de vues matérialisées incohérentes, telle que la vue combinant le paramètre *class* caractérisant les véhicules américains et le paramètre *Département* caractérisant une agence française. A notre connaissance, ce problème n'a été traité par aucun des travaux antérieurs.

La taille du treillis ***T*** dépend du nombre d'attributs dans les dimensions, et plus particulièrement dans les hiérarchies. Sans considération des contraintes :

$$T = \Pi_{i=1}^{h} n_i$$

Où n_i est le nombre d'attributs dans la hiérarchie i et h le nombre de hiérarchies combinées dans le treillis.

L'intégration des contraintes réduit la taille du treillis en enlevant les combinaisons incohérentes.

Dans la section suivante, nous présentons les étapes de construction du treillis dimensionnel en intégrant les différentes contraintes sémantiques définies dans le modèle dimensionnel.

3.2. Construction du treillis dimensionnel

Dans cette section, nous présentons notre processus de construction du treillis dimensionnel contraint représentant l'ensemble des vues à matérialiser. Dans un premier temps, nous proposons de construire un premier treillis dimensionnel qui intègre la structure des hiérarchies (Harinarayan et al, 1996). Dans un second temps, nous intégrons les contraintes sémantiques dans le processus de construction de ce treillis. Puis, nous validons l'apport de ces contraintes permettant de réduire le nombre de vues à matérialiser.

3.2.1. Construction du treillis dimensionnel sans contraintes

Soit V l'ensemble des vues possibles résultant de la combinaison des paramètres des dimensions d'un fait. Pour chaque fait de notre constellation, nous pouvons définir un treillis dimensionnel de vues $v \in V$ de type ***ET-OU***. Le choix de ce type de treillis permet d'exprimer la sémantique des contraintes intra et inter-dimensions. En effet, l'existence d'une contrainte d'exclusion ou de partition entre deux hiérarchies, dont les paramètres sont combinés dans deux vues différentes du treillis, implique que ces vues ne couvrent pas la totalité des données analysées. C'est le lien de type ***ET*** qui permet de représenter ce fait ; il permet d'indiquer que le calcul d'une vue se fait à partir des données de plusieurs vues à la fois.

Nous proposons de construire le treillis en se basant sur la structure hiérarchique (Harinarayan et al, 1996) afin d'inclure la sémantique des hiérarchies.

La taille du treillis dimensionnel est réduite en se basant sur la notion de dépendance fonctionnelle (*DF*) entre les paramètres d'une hiérarchie. En effet, si nous considérons deux attributs a_i et a_j et que a_j dépend fonctionnellement de a_i, notée $a_i \rightarrow a_j$, (a_i détermine a_j) alors le regroupement des données selon a_i donne le même résultat que le regroupement selon le couple (a_i, a_j). Par exemple, regrouper les locations par *Région* et par *Pays* donne le même résultat qu'un regroupement par *Région* puisque une *région* détermine le *pays*. Dans le treillis, seul le nœud regroupant les ventes par *région* est représenté.

En se basant sur cette notion de dépendance, nous définissons un nœud de notre treillis dimensionnel comme suit.

Définition

Un nœud N^P appartenant au treillis dimensionnel, représentant une vue v, est défini par l'ensemble de ses paramètres P comportant les attributs de regroupement de la vue v et tel qu'il n'existe pas de dépendances fonctionnelles entre les paramètres de P.

En respectant cette définition de nœud, la construction du treillis est réalisée d'une manière récursive :

1. L'algorithme part de la vue la plus détaillée comportant comme paramètres les identifiants des dimensions analysées (cette vue représente la racine du treillis).
2. Chaque paramètre de cette vue est remplacé par l'ensemble des paramètres qui en dépendent fonctionnellement (ligne 3, Figure III.25). Dans notre exemple, le paramètre *CodeAg* est remplacé par le paramètre *Ville*. Ce dernier est remplacé, ensuite, par les paramètres *Zone*, *Département* et *Etat* afin de former la combinaison (*Zone, Département, Etat*).
3. L'ensemble des paramètres obtenu est épuré (ligne 4, Figure III.25) selon la méthode suivante : tout paramètre dépendant fonctionnellement d'un autre est éliminé de l'ensemble.
4. Ce nouvel ensemble de paramètres constitue les paramètres de regroupement de la nouvelle vue. Cette dernière est ajoutée au treillis si elle n'existe pas déjà.
5. L'appel récursif de l'algorithme de construction (ligne 9, Figure III.25) est réalisé en se basant sur les paramètres de la nouvelle vue créée. L'algorithme revient alors à l'étape 2. et ainsi de suite jusqu'à la construction de tout le treillis.

Exemple 16

Dans notre exemple de la Figure III.26 (a), la vue la plus détaillée représente les locations en fonction du paramètre *CodeAg*. A partir de cette vue, nous construisons le treillis dimensionnel de la Figure III.26 (b) étape par étape comme suit :

Appel récursif N° 1
1. R = {(CodeAg)}
2. CodeAg ➔ Ville
3. le nœud Ville est épuré
4. T = {(CodeAg), (Ville)}
5. ***ConstructionTreillis***(Ville)

Appel récursif N° 2
1. R = {(Ville)}
2. Ville ➔ *Zone, Département, Etat*
3. le nœud (*Zone, Département, Etat)* est épuré.
4. T = {(CodeAg), (Ville), (*Zone, Département, Etat)*}
5. ConstructionTreillis((*Zone, Département, Etat)*)

Appel récursif N° 3
1. R = {(*Zone, Département, Etat)*}
2. Zone ➔ *Pays ; On obtient (Pays, Département, Etat)*
3. le nœud (*Pays, Département, Etat)* est épuré ➔ (*Département, Etat)*
4. T = {(CodeAg), (Ville), (*Zone, Département, Etat), (Département, Etat)*}
5. ConstructionTreillis((*Département, Etat)*)

…

Dans cet exemple, nous présentons le treillis comme un ensemble de nœuds (T) en simplifiant les liens entre ces nœuds. Par exemple, le nœud (*Zone, Département, Etat)* aura comme nœud père le nœud (*Ville*) à partir duquel il a été calculé, et comme nœud fils le nœud (*Département, Etat*).

Algorithme ConstructionTreillis
`Entrée :`
Ax : un nœud du treillis Avec :
Structure Nœud :
Racine : liste des paramètres,
LFils = liste des fils du nœud.
LPere_ET : pères reliés par un arc ET.
ListeNoeud : Liste des nœuds construits.
`Sortie :` *Liste des nœuds du treillis construit*
`Début`
1. *Pour chaque Pi ∈ Ax.Racine Faire*
2. *Lp ← Ax.Racine - {Pi};*
3. *Lp ← Lp +* ***Param*** *(p_i) ;*
4. *Lp ←* ***Epurer*** *(Lp) ;*
5. *Az ←* ***ChercheNoeud*** *(Lp, ListeNœud)*
6. *Si (Az= nul) Alors*
7. *Az.Racine ← Lp ;*
8. *ListeNœud ← ListeNœud + Az ;*
9. *ListeNœud ←* ***ConstructionTreillis*** *(Az, ListeNœud) ; //appel récursif*
10. *FinSi*
11. *Ax.LFils ← Ax.LFils +Az ;*
12. *FinPour*
13. *Retourner (ListeNœud)*

FinConstructionTreillis
Algorithme Epurer
`Entrée :` *Liste de paramètres.*
`Sortie :` *Liste de paramètres épurés.*
`Début`
1. *Pour chaque Pi ∈ Lp Faire*
2. *Pour chaque Pj ∈ Lp- Pi Faire*
3. *Si (Pj ∈* ***Dépend****(Pi)) Alors Lp ← Lp – Pi ;*
4. *Finfaire*
5. *FinPour*
6. *Retourner (Lp) ;*

FinEpurer
Param *(p) récupère la partie droite (a) des dépendances fonctionnelles p → a.*
ChercheNoeud *(Lp, ListeNœud) Cherche le nœud de racine Lp dans ListeNœud*
Dépend *(p_i) récupère les paramètres qui dépendent fonctionnellement de p_i*

Figure III.25 *: Algorithme de création d'un treillis sans contraintes*

Exemple 17

L'application de ce premier algorithme sur les paramètres de la dimension *Agences,* nous permet de construire le treillis de la Figure III.26 (b).

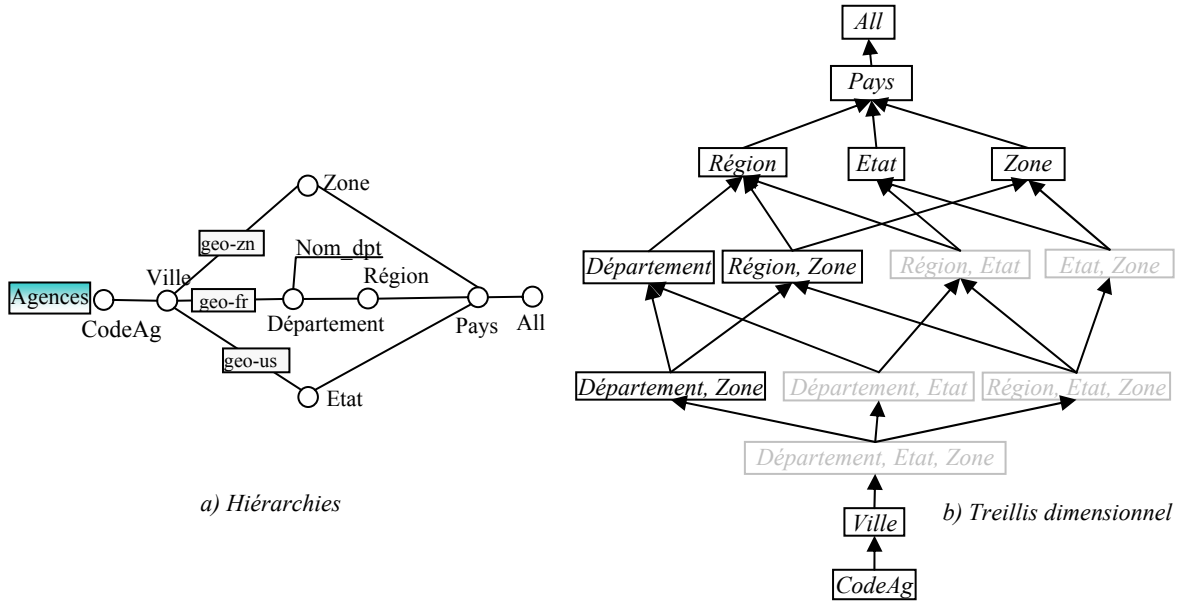


***Figure III.26** : Treillis des locations selon la dimension Agences (Baralis et al, 1997)*

L'exemple présenté combine les hiérarchies d'une seule dimension afin de faciliter la compréhension du processus de construction du treillis. Le même processus est appliqué pour combiner un ensemble de dimensions.

Dans ce treillis, la combinaison des différents paramètres réalisant les vues ne tient pas compte des contraintes entre les hiérarchies. La vue visualisant les montants des locations en fonction des paramètres *Département* et *Etat* et celle combinant les paramètres *Région* et *Etat* ne peuvent pas exister car une contrainte d'exclusion est définie entre la hiérarchie de la géographie française et celle de la géographie américaine. Dans la Figure III.26 (b), nous avons grisé les vues qui violent l'intégrité des contraintes exprimées dans le modèle.

En outre, la vue qui décrit les montants des locations pour chaque pays peut être extraite à partir des vues matérialisées regroupant les montants des locations par région, état ou zone géographique. Sans considération des contraintes entre les instances des hiérarchies, nous n'aurons pas d'information sur la complétude de cette vue. En effet, regrouper les montants des locations par *Pays* à partir du paramètre *Région* ne donne qu'une partie des instances de la dimension, puisque le paramètre *Région* ne concerne que les ventes réalisées en France. Par contre, le passage du paramètre *Zone* vers *Pays* permet de calculer les montants des locations pour toutes les instances de la dimension *Agences*. L'introduction des contraintes sur les instances de la dimension permet d'enlever cette ambiguïté.

3.2.2. Intégration des contraintes

Pour intégrer les contraintes, nous modifions l'algorithme de construction de treillis en validant chaque nœud avant de l'insérer dans le treillis. La validation est basée sur les contraintes définies sur les instances des faits et des dimensions.

Définition

Un Nœud N^P est Valide en considérant l'ensemble des contraintes C, s'il n'existe aucun couple de paramètres (p_i, p_j) appartenant à P x P qui sont en exclusion.

Définition

Deux paramètres p_i et $p_j \in P$ sont en exclusion si toutes les hiérarchies h_n passant par p_i sont en exclusion (ou en partition) avec toutes les hiérarchies h_m passant par p_j.

```
Algorithme ValiderNœud
Entrée : Un nœud Ax
Sortie : Vrai si le nœud est valide, sinon faux.
Début
    1.  Lp ← Ax.Racine ;
    2.  Valide ← Vrai ;
    3.  i ← 0 ;
    4.  Tant Que Valide et i< taille (Lp) Faire
    5.      Pi ← Lp[i];
    6.      j ← i+1 ;
    7.      Tant Que (Valide et j< taille (Lp)) Faire
    8.          Pj ← Lp[j] ;
    9.          Valide ←ExclusionP (pi, pj);
    10.     FinTQ
    11. FinTQ
    12. Retourner (Valide) ;
FinValiderNœud

Algorithme ExclusionP
Entrée : Deux paramètres pi et pj
Sortie : Vrai si Pi et Pj sont en exclusion, sinon faux.
Début
    1.  Lhi ← les hiérarchies passant par pi ;
    2.  Lhj ← les hiérarchies passant par pj ;
    3.  p ← 0 ;
    4.  Valide = Vrai ;
    5.  Tant Que non Valide et p< taille (Lhi) Faire
    6.      hp ← Lhi [p] ;  q ← 0 ;
    7.      Tant Que non Valide et q <taille (Lhj) Faire
    8.          hq ← Lhj [q] ;
    9.          Si ((Exclusion (hq, hp)) = Faux) Alors Valide ← Faux;
    10.     FinTQ
    11. FinTQ
    12. Retourner (Valide) ;
FinExclusionP
```

***Figure III.27** : Algorithme de validation des contraintes des nœuds du treillis*

Ce premier algorithme de simplification (Figure III.27) permet d'enlever du treillis les nœuds représentant des vues qui ne respectent pas les contraintes de validité. L'appel à la procédure "*ExclusionP (Pi, Pj)*" (ligne 11) permet de vérifier si les paramètres d'un nœud, pris deux à deux, sont en exclusion (définition 7).

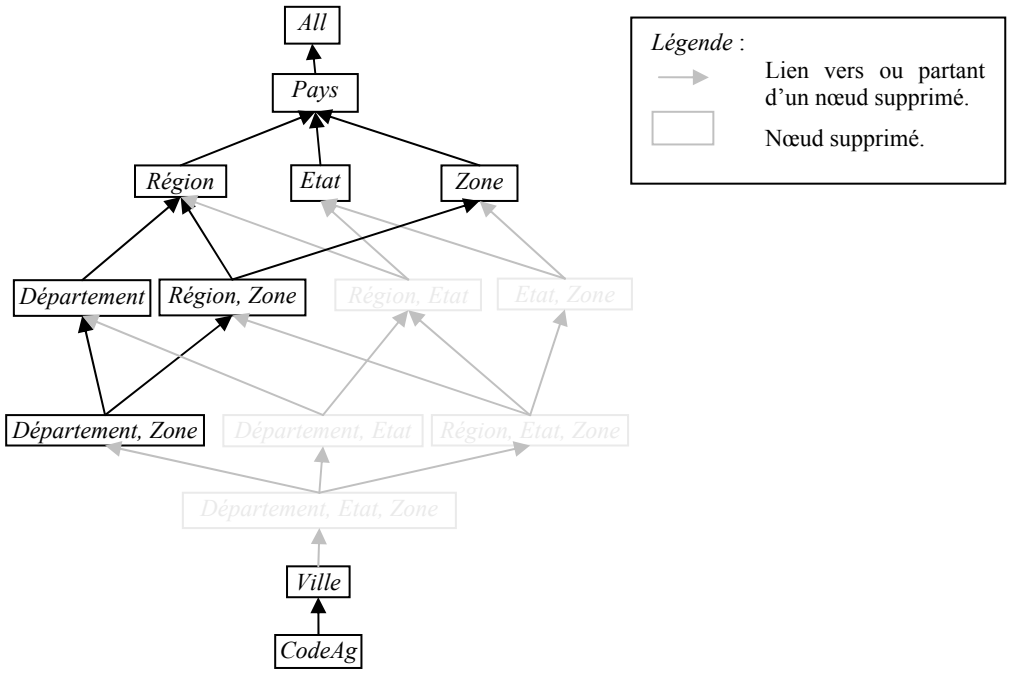


***Figure III.28** : Suppression des nœuds invalides du Treillis de la dimension Agences*

Dans l'exemple que nous avons présenté en Figure III.26 (a), nous proposons de modifier le treillis afin de tenir compte des contraintes sur les instances de la dimension *Agences*. Un nœud supprimé à la suite de cette modification laisse une place vide dans le treillis vers laquelle pointe les autres nœuds (les nœuds supprimés sont présentés en gris clair dans la Figure III.28).

Pour obtenir le treillis valide, il faut enlever les liens vers les nœuds fils supprimés du treillis et les remplacer par des liens qui pointent vers des nœuds fils valides.

Algorithme ValiderLiens
Entrée : *Liste des nœuds du treillis*
Sortie : *Treillis avec des liens valides.*
Début
1. *Pour chaque nœud Ax ∈ ListeNœud Faire*
2. *Pour chaque nœud Af ∈ Ax.LFils Faire*
3. *Si Af ∉ ListeNœud Alors*
4. *Ax.LFils ← Ax.LFils – Af ;*
5. *Ax.LFils ← Ax.LFils + Af.LFils;*
6. *FinSi*
7. *FinPour*
8. *FinPour*
9. *Pour chaque nœud Ax ∈ ListeNœud Faire*
10. *Pour chaque Af1 ∈ Ax.Lfils Faire*
11. *Pour chaque Af2 ∈ Ax.LFils – Af1 Faire*
18. *Si Af2 ∈* ***petitFils(****Af1) Alors Ax.LFils ← Ax.LFils – Af2;*
19. *Si Af1 ∈* ***petitFils(****Af2****)*** *Alors Ax.LFils ← Ax.LFils – Af1;i*
20. *FinPour*
21. *FinPour*
22. *//****Construction des arcs ET****.*
23. *Pour chaque Ap1 ∈* ***pere****(Ax) Faire*
24. *Pour chaque Ap2 ∈* ***pere*** *(Ax) – Ap1 Faire*
25. *Si (****ExclusionN (****Ap1, Ap2****))*** *Alors Ajouter {Ap1, Ap2} à Ax.LPere_ET;*
26. *FinPour*
27. *FinPour*
28. *FinPour*

FinValiderLiens

Algorithme ExclusionN
Entrée : *Deux nœuds Ni et Nj.*
Sortie : *Vrai si Ni et Nj sont en exclusion, sinon faux.*
Début
1. *Li← Ni.Racine ;*
2. *Lj← Nj.Racine ;*
3. *Valide ← Vrai ;*
4. *i←0 ;*
5. *Tant Que Valide et i< taille(Li) Faire*
6. *Pi ← Li[i] ;*
7. *j ← 0 ;*
8. *Tant Que Valide et j< taille (Lj) Faire*
9. *Pj ← Lj[j];*
10. *Valide ←* ***ExclusionP (****p_i, p_j****);*
11. *FinTQ*
12. *FinTQ*
13. *Retourner (Valide) ;*

FinExclusionN
petitFils (Af) *: récupère les petits fils du nœud Af dans le treillis.*
pere (Ax) *: récupère les nœuds pères de Ax.*

Figure III.29 *: Algorithme de reconstruction des liens intégrant les contraintes*

Le deuxième algorithme (Figure III.29) permet de reconstruire les liens entre les nœuds valides du treillis dimensionnel et d'ajouter les arcs ET. La Figure III.30 décrit les différentes étapes de reconstruction des liens par cet algorithme.

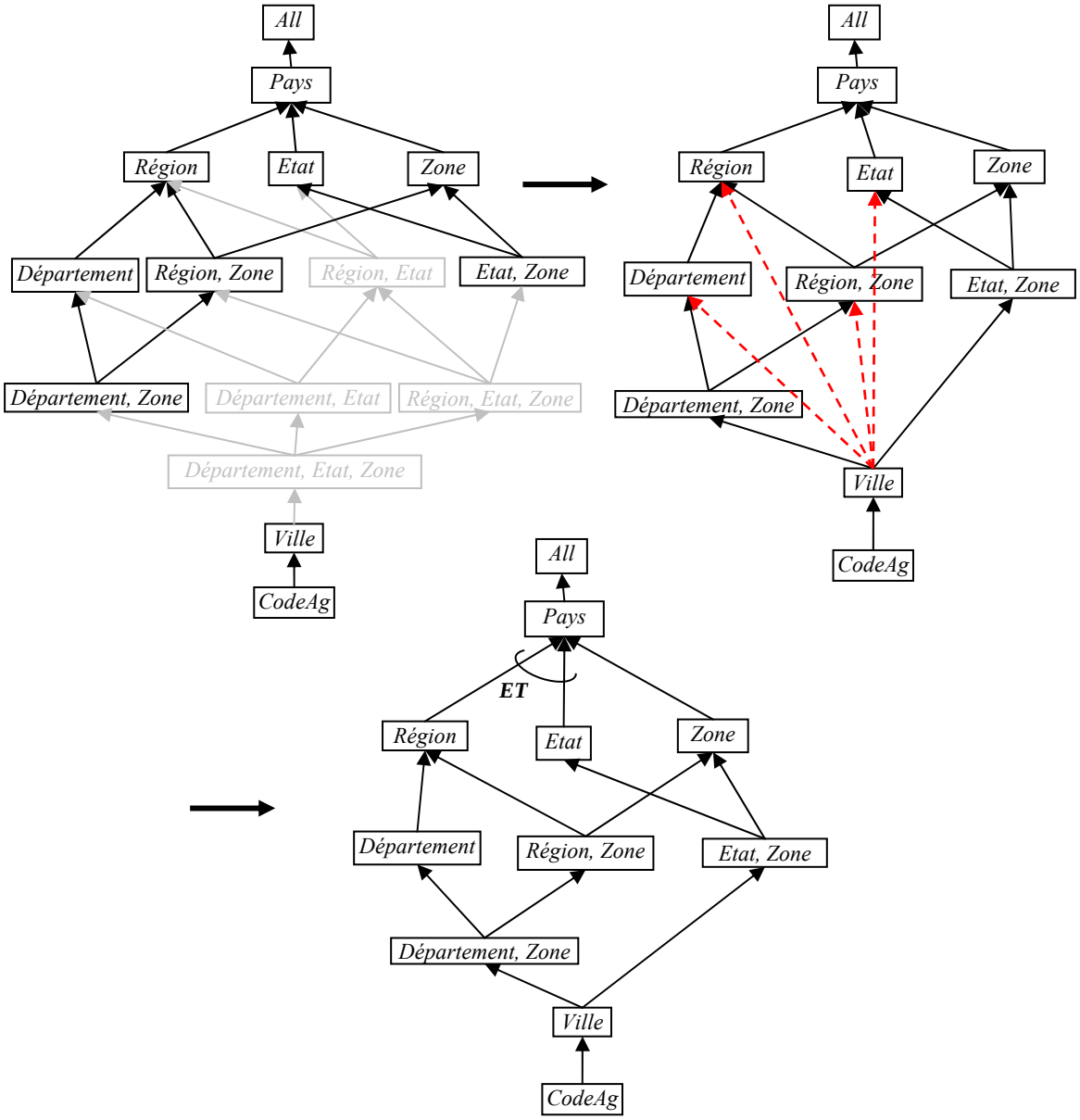


Figure III.30 *: Treillis de la dimension Agences intégrant les contraintes*

La vue regroupant les locations par pays est calculée soit à partir de la vue regroupant les locations par *Zone* soit à partir de l'union des montants des locations par *Etat* et par *Région*. La notation '*ET'* indique que l'union des vues comportant les montants des locations par région et celle comportant les montants des locations par état, comporte la totalité des locations de toutes les agences dans tous les pays. Ce fait est exprimé par la contrainte de partition définie entre les hiérarchies "*geo_fr*" et "*geo_us*".

En outre, cette contrainte, intégrée dans le treillis, interdit de combiner les deux paramètres *Etat* et *Département* en une seule vue (de même pour *Etat* et *Région*). La combinaison entre les paramètres *Département* et *Zone*, d'un coté, et *Etat* et *Zone*, de l'autre, est permise donnant lieu à deux nouvelles vues possibles pour les ventes.

Le traitement que nous avons réalisé au niveau d'une seule dimension est applicable au niveau d'un fait entre les hiérarchies de dimensions différentes. Dans notre exemple (Figure III.5), deux contraintes d'exclusion sont définies entre les hiérarchies "*clas_fr*" et "*geo_us*", d'une part, et les hiérarchies "*clas_us*" et "*geo_fr*", d'autre part.

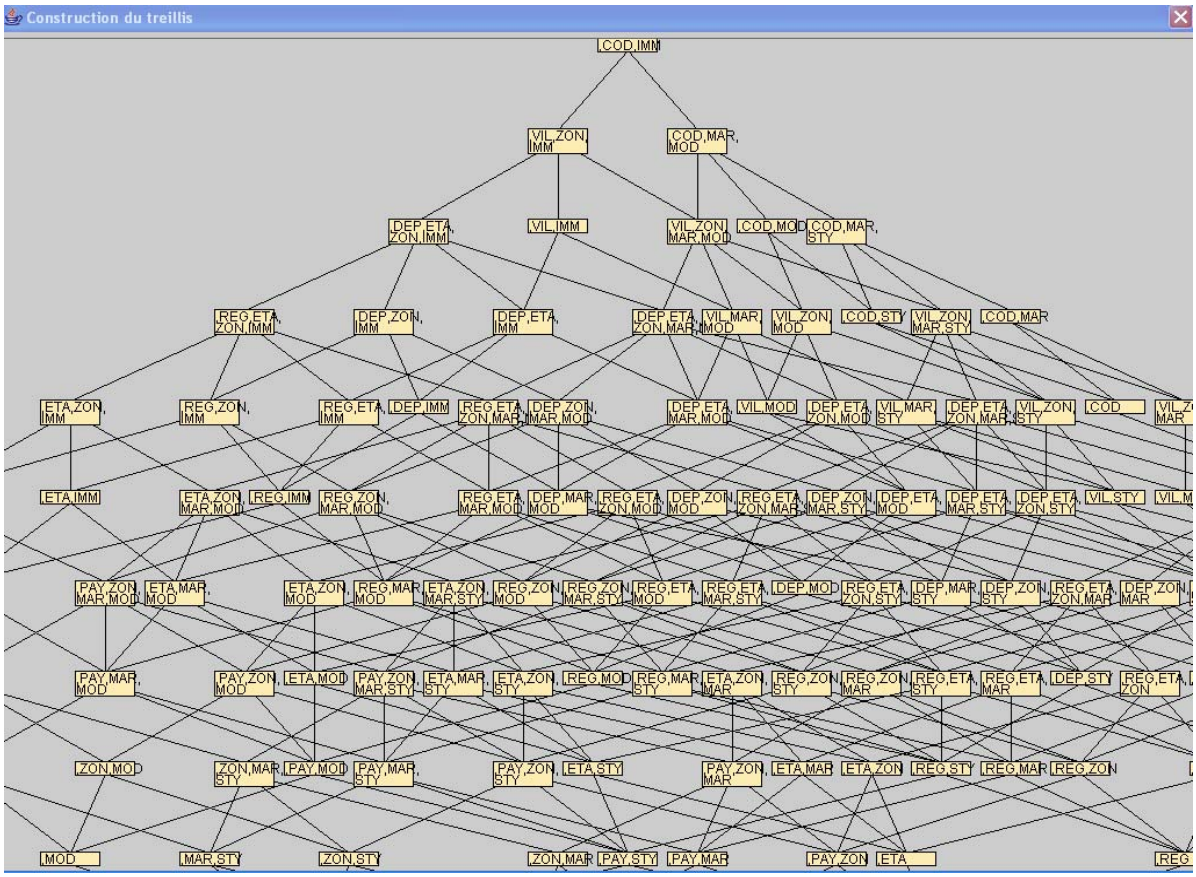
3.2.3. Validation

Afin de valider notre proposition, nous avons implanté l'algorithme de construction du treillis dimensionnel et nous l'avons appliqué sur un exemple de base de données en calculant le nombre de vues dans les treillis obtenus avant et après intégration des contraintes sémantiques. Pour simplifier, nous avons pris l'exemple de la BDM exposée dans la Figure III.3. Dans cet exemple, nous avons construit les treillis dimensionnels en combinant les différentes dimensions sans intégrer les contraintes afin de relever le nombre de vues de chaque treillis. Puis, nous avons intégré les contraintes dans la base et nous avons construit de nouveau l'ensemble des treillis possibles (voir Tableau III.2).

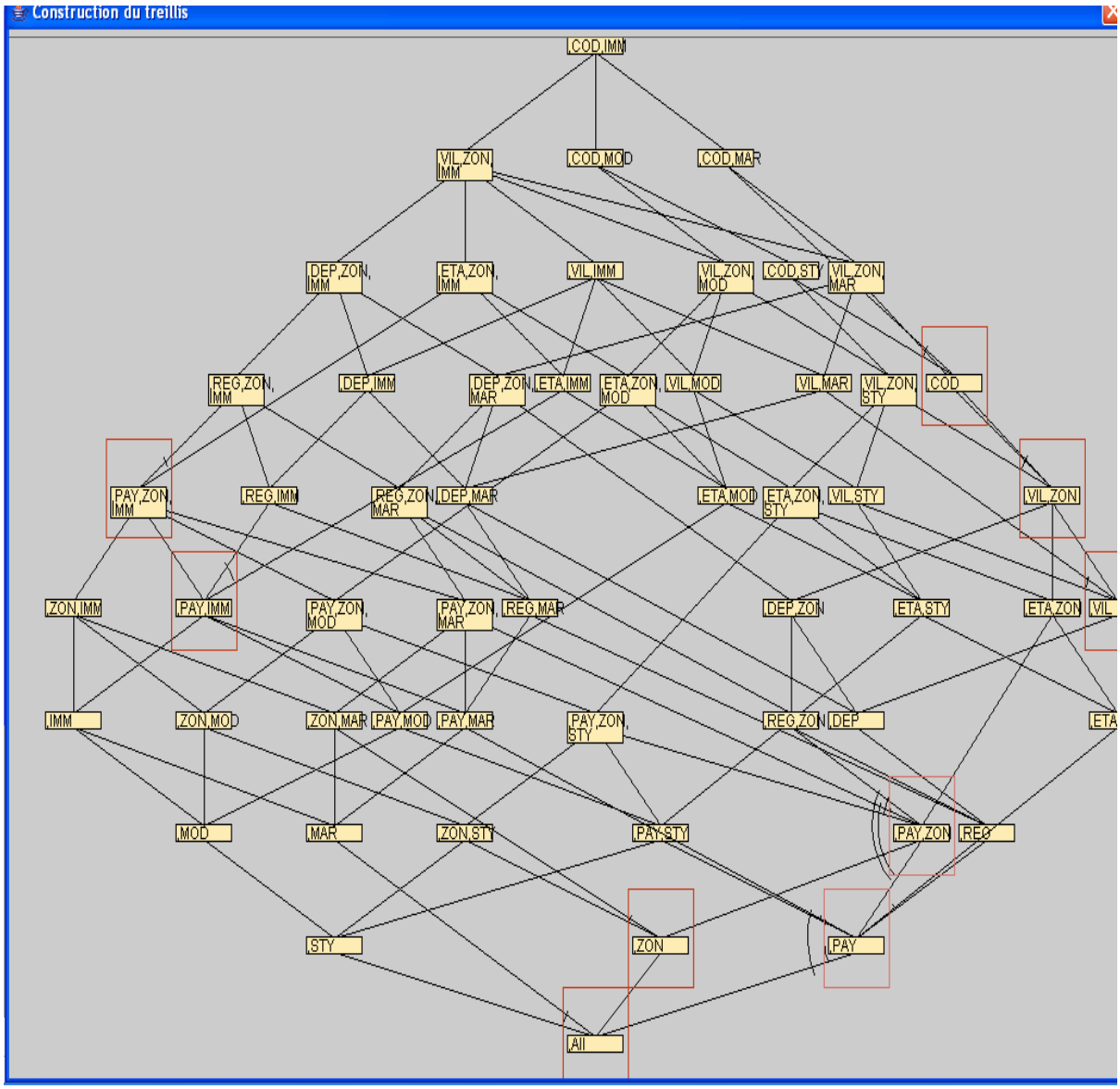
Taille du treillis dimensionnel	**Véhicule**	**Agence**	**TEMPS**	**Véhicule Agence**	**Véhicule, TEMPS**	**Agence, TEMPS**	**Véhicule, Agence TEMPS**
Sans contraintes	7	14	5	119	35	85	595
Avec contraintes	5	10	5	55	25	65	275

Tableau III.2 *: Tableau comparatif de la taille du treillis multidimensionnel.*

La diminution du nombre de vues dans le treillis dimensionnel simplifie sa manipulation. La recherche d'un nœud représentant une vue à matérialiser devient moins coûteuse et plus facile. Si nous reprenons l'exemple du treillis combinant les ventes selon les dimensions *Véhicules* et *Agences*, nous passerons d'un treillis de 119 nœuds sans considérer les contraintes (Figure III.31 (a)) vers un treillis de 55 nœuds après intégration des contraintes sémantiques (Figure III.31 (b)).



(a) avant intégration des contraintes



(b) après intégration des contraintes

Figure III.31 *: Treillis combinant les dimensions Véhicules et Agences*

4. Conclusion

Dans ce chapitre, nous avons étudié l'impact des contraintes sur les opérations dimensionnelles. Aspect qui est peu étudié dans les travaux existants. Afin de tenir compte des contraintes définies dans notre modèle de données en constellation, nous avons proposé des opérateurs dimensionnels qui intègrent une nouvelle propriété offrant la possibilité de maintenir ou d'étendre les analyses en fonction des contraintes existantes entre les hiérarchies. Ainsi, nous avons proposé deux opérateurs de visualisation : libre et en fonction des hiérarchies qui permettent d'afficher une première table dimensionnelle répondant aux besoins des décideurs. Les décideurs pourront, par la suite, naviguer dans les données à l'aide des fonctions de forage qui fixent l'ensemble des instances à analyser. Nous proposons également, des opérateurs de rotations de perspectives et d'axes d'analyse qui tiennent compte des contraintes entre les hiérarchies. Enfin, nous avons étudié l'impact des contraintes

sur les opérateurs d'emboîtement. Notre langage d'interrogation des données dimensionnelles contribue à améliorer la cohérence des analyses :

- en validant les données restituées par rapport aux contraintes, et
- en informant le décideur du changement de l'ensemble des données analysées causé par les contraintes (Ghozzi et al, 2003b) (Ghozzi et al, 2004).

Un autre objectif de nos travaux consiste à exploiter les contraintes exprimées dans le modèle afin d'optimiser l'interrogation en diminuant le temps de réponse aux requêtes des décideurs. Notre approche consiste à améliorer le processus de sélection des vues matérialisées en se basant sur le concept de treillis dimensionnel. Nous avons commencé par définir un treillis dimensionnel qui se base sur la structure des hiérarchies. Dans une seconde étape, nous avons réalisé l'intégration des contraintes dans le processus de construction des treillis permettant d'enlever toutes les vues inadéquates ne satisfaisant pas les contraintes et de diminuer, ainsi, le nombre des vues à matérialiser (Ghozzi, 2003a) (Ghozzi, 2003b). En outre, l'intégration de la sémantique des contraintes dans le treillis contribue à la définition des liens entre les vues du treillis (Lien ET-OU). Ces liens permettent de calculer correctement une vue à partir d'autres vues dans le treillis dimensionnel.

Dans ce chapitre, nous nous sommes focalisés sur la manipulation des données dimensionnelles contraintes. Néanmoins, la conception d'une base de données dimensionnelles nécessite la définition d'une méthode rigoureuse afin de modéliser correctement les besoins décisionnels en tenant compte des contraintes sémantiques. Cette problématique fait l'objet du développement du chapitre IV.

CHAPITRE IV : METHODE DE CONCEPTION D'UN SCHEMA DIMENSIONNEL

PLAN DU CHAPITRE

L'objet de ce chapitre est la proposition d'une méthode de conception de schéma dimensionnel intégrant l'expression d'un ensemble de contraintes sémantiques. En effet, la conception d'une base décisionnelle est une tâche critique et difficile à mettre en oeuvre. Elle doit répondre de façon spécifique aux besoins de l'entreprise et apporter les bonnes réponses aux requêtes des décideurs. Aussi, la définition d'une **méthode de conception** d'une telle base est cruciale afin de mener à bien cette tâche (Kimball et al, 2002).

Dans la première section de ce chapitre, nous présentons la problématique générale liée à la proposition d'une méthode de conception de magasin de données dimensionnelles. La deuxième section présente le modèle d'entrepôt de données à partir duquel nous concevons notre schéma dimensionnel. Dans la troisième section, nous décrivons l'architecture de notre méthode de conception qui comporte trois étapes : une démarche descendante, une démarche ascendante et une étape de confrontation. La description de ces trois étapes fait l'objet des sections quatre, cinq et six.

1. Introduction

Peu de méthodes de conception sont proposées dans le domaine de la modélisation dimensionnelle (Trujillo et al, 2003). Dans la littérature, nous distinguons trois catégories de méthodes de conception de schémas dimensionnels.

- les méthodes ascendantes qui utilisent les sources de données pour définir les besoins des décideurs et pour concevoir les schémas dimensionnels (List et al, 2002) (Moody et al, 2000). Ces méthodes ne tiennent pas compte des besoins des décideurs ;
- les méthodes descendantes qui commencent par l'analyse et la définition des besoins utilisateurs. Les données des sources ne sont pas prises en compte (Kimball et al, 2002) ;
- les méthodes mixtes qui se basent sur les données sources pour définir les schémas dimensionnels en y intégrant les besoins des utilisateurs (Trujillo et al, 2003).

Les méthodes de conception proposées se basent, pour la plupart, sur un modèle dimensionnel logique ou conceptuel (Trujillo et al, 2002) (Moody et al, 2000). Aucun de ces modèles n'expriment les contraintes entre les données dimensionnelles. Or, ces contraintes permettent une définition des schémas dimensionnels interdisant des incohérences dans les analyses (Hurtado et al, 2002) et une meilleure prise de décision en offrant une information fiable aux décideurs.

Afin de palier ces insuffisances, nous proposons une méthode de conception des bases de données dimensionnelles intégrant l'expression des contraintes. Celle que nous proposons est basée sur une approche mixte intégrant à la fois la description des données sources et la définition des besoins utilisateurs. Ce choix est motivé par le fait que ce type de méthode permet d'intégrer toutes les données pertinentes pour la prise de décision (Trujillo et al, 2003). De plus, contrairement à plusieurs méthodes proposées, nous définissons une méthode de conception complète qui comporte un ensemble de concepts défini dans notre modèle dimensionnel contraint, un ensemble de formalismes graphiques représentant ces concepts, des démarches de conception de schéma dimensionnel et un outil d'aide à la conception de tel schéma.

Les bases dimensionnelles représentent une composante de notre architecture de système décisionnel. Nous rappelons l'architecture générale de notre système décisionnel à la Figure IV.1 (Ravat et al, 2000a).

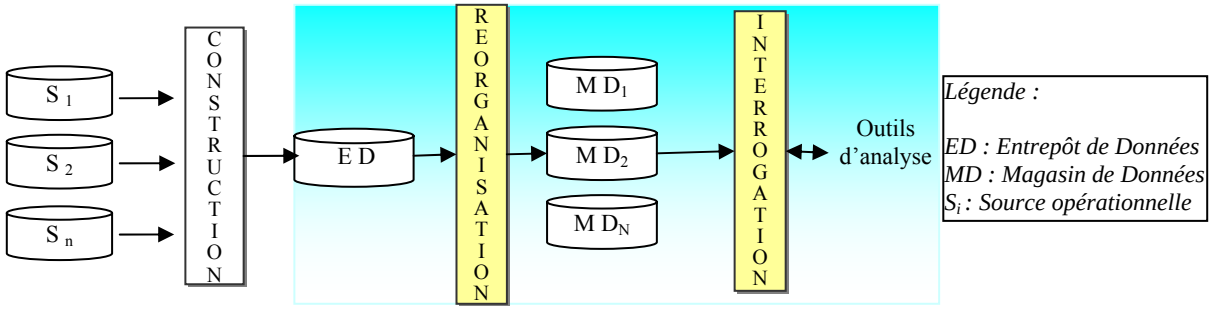


Figure IV.1 : *Entrepôt et magasins de données.*

Cette architecture adopte une dichotomie d'espace de stockage : l'entrepôt et les magasins de données. Dans ce cadre, nous proposons une méthode de conception de magasin de données dimensionnelles à partir d'un entrepôt de données historisées. Afin de pouvoir détailler notre méthode de conception, la deuxième section de ce chapitre présente brièvement le modèle de l'entrepôt proposé par notre équipe (Teste, 2000).

Dans les sections suivantes, nous présentons notre méthode de conception de schéma dimensionnel. Dans cette méthode, nous suivons en parallèle une démarche descendante basée sur la spécification des besoins des décideurs et une démarche ascendante basée sur la description des données de l'entrepôt. Les schémas dimensionnels résultants de ces deux démarches sont confrontés dans une dernière étape pour produire un schéma conceptuel dimensionnel contraint.

2. Modèle de l'entrepôt

Au sein de notre équipe, nous avons proposé un modèle conceptuel orienté objet pour la représentation des données de l'entrepôt qui intègre des données temporelles et archivées (Teste, 2000). Le choix de l'orienté objet est justifié par le fait que le modèle objet permet de présenter facilement les différents types de données existants dans les différentes sources de production (Bukhres et al, 1993). En outre, l'entrepôt doit permettre de gérer les données temporelles nécessaires à la prise de décision (Yang et al, 2000) (Pedersen et al, 1999). Aussi, le modèle de l'entrepôt est un modèle orienté objet temporel dédié à la prise de décision.

Ce modèle se base sur le langage de modélisation objet UML. Nous présentons dans la section suivante, les concepts de base du modèle d'entrepôt défini au sein de notre équipe. Ce modèle étend la définition de classes UML et introduit le concept d'environnement afin de supporter l'historique des données (détaillé et archivé).

2.1. Concept d'objet entrepôt

Au niveau de l'entrepôt, chaque objet source extrait (ou groupe d'objets source) est représenté par un objet, appelé objet entrepôt. L'entrepôt de données peut conserver les changements d'états des objets, tandis que la source de données ne contient que l'état courant

ou bien une partie récente des évolutions, insuffisante pour la prise de décision (Chaudhuri et al, 1997) (Yang et al, 2000). Dans un entrepôt, l'administrateur peut décider de conserver :

- uniquement, l'image source, c'est-à-dire son **état courant**,
- les états successifs que prend l'objet source dans le temps, appelés **états passés,**
- un résumé de ses états passés successifs, c'est-à-dire l'agrégation de certains états passés en **états archivés**.

La Figure IV.2 décrit le principe de la modélisation des objets entrepôt, en représentant un objet entrepôt possédant un état courant, deux états passés et un état archivé.

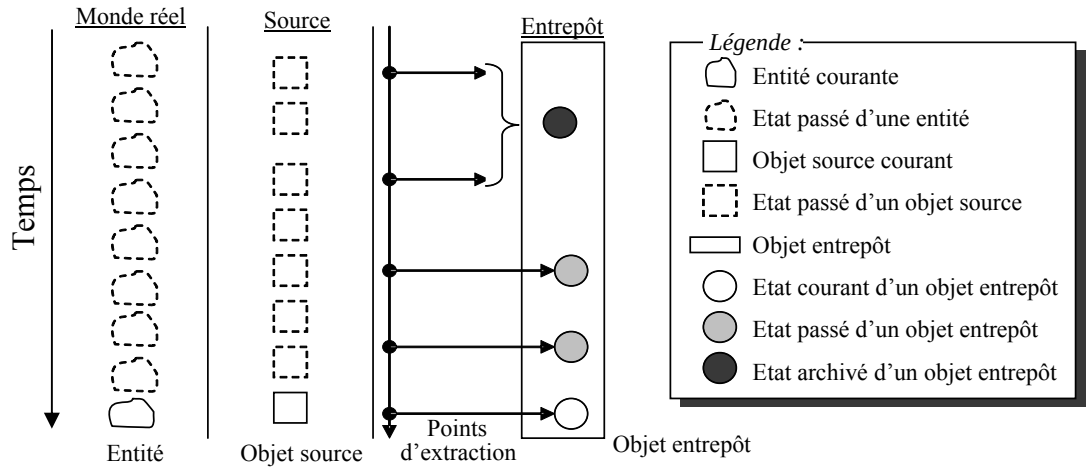


Figure IV.2 : *Principe de modélisation d'un objet entrepôt*

La mise à jour de l'entrepôt est réalisée de manière périodique. Autrement dit, elle s'effectue à un **point d'extraction** correspondant à un instant où la base de données source est dans un état cohérent (fin des transactions de mise à jour). Ainsi, l'état d'un objet entrepôt ne correspond pas forcément à l'état courant de la source et certaines évolutions de la source ne sont pas immédiatement répercutées dans l'entrepôt. L'administrateur doit équilibrer l'exigence de performance de l'entrepôt de données tout en garantissant un rythme (période) de rafraîchissement suffisant pour ne pas perdre des évolutions utiles de la source.

Nous formalisons le concept d'objet entrepôt par les définitions suivantes :

Définition

*Un **objet entrepôt** O est défini par le quadruplet (Oid, S^0, E^P, E^A) où :*

- *Oid est l'identifiant interne,*
- *S^0 est l'état courant,*
- *$E^P = \{S^p_1, S^p_2, \ldots S^p_n\}$ est un ensemble fini d'états passés,*
- *$E^A = \{S^a_1, S^a_2, \ldots S^a_m\}$ est un ensemble fini d'états archivés.*

Définition

Un état S_i d'un objet entrepôt est défini par le couple (h_i, v_i) où :

- *h_i est le domaine temporel correspondant aux instants durant lesquels l'état S_i est courant,*
- *v_i est la valeur de l'objet durant les instants de h_i.*

2.2. Concept de classe entrepôt

Le concept de classe entrepôt étend le concept standard de classe défini dans UML afin d'intégrer le caractère évolutif des objets entrepôt et de caractériser le processus de construction par extraction.

Définition

*Une **classe entrepôt c** est définie par un n-uplet (Nom^c, $Type^c$, $Super^c$, $Extension^c$, $Mapping^c$, $Tempo^c$, $Archi^c$) où*

- *Nom^c est le nom de la classe,*
- *$Type^c$ est le type de la classe ; il définit la structure et le comportement des objets de la classe,*
- *$Super^c$ est l'ensemble des super-classes de c,*
- *$Extension^c$ est l'ensemble fini d'objets entrepôt regroupés dans la classe c,*
- *$Mapping^c$ est **la fonction de construction** qui caractérise le processus d'extraction à partir duquel la classe c est générée et alimentée en instances,*
- *$Tempo^c$ est **le filtre temporel** qui caractérise l'ensemble des propriétés temporelles de la classe décrivant des données pertinentes pour le système décisionnel. Il est constitué d'attributs temporels dont les évolutions de valeurs sont conservées par des états passés,*
- *$Archi^c$ est **le filtre d'archives** qui caractérise l'ensemble des attributs archivés de la classe dont les évolutions sont conservées de manière agrégée. Le filtre d'archives est un ensemble de couples (attribut, fonction) ; l'ensemble des attributs archivés est associé à une fonction d'agrégation qui indique comment sont résumées les évolutions détaillées dans les états d'archives.*

2.3. Concept d'environnement

Dans un entrepôt de données, les classes n'ont pas toutes le même comportement temporel. Ainsi, certaines classes sont mises à jour quotidiennement et d'autres à un rythme mensuel. Le concept d'environnement (Ravat et al, 2000b) permet de modéliser cette réalité en définissant des parties temporelles homogènes, cohérentes et configurables dans l'entrepôt.

Définition

*Un **environnement** Env est défini par le triplet (Nom^{Env}, C^{Env}, $Config^{Env}$) où :*

- *Nom^{Env} est le nom identifiant l'environnement,*
- *$C^{Env} = \{c_1, c_2, \ldots, c_m\}$ est l'ensemble fini des classes contenues dans l'environnement,*
- *$Config^{Env}$ est un ensemble de règles de configuration, visant à définir différents paramètres locaux à l'environnement (période de rafraîchissement, …).*

Toute classe historisée doit appartenir à un environnement qui détermine, à l'aide de sa configuration, les périodes de rafraîchissement et d'archivage des données.

2.4. Concept d'entrepôt

Définition

*Un **entrepôt** ED est défini par le n-uplet (Nom^{ED}, C^{ED}, Env^{ED}, $Config^{ED}$) où :*

- *Nom^{ED} est le nom de l'entrepôt,*
- *$C^{ED} = \{c_1, c_2, \ldots, c_n\}$ est l'ensemble fini des classes de l'entrepôt,*
- *$Env^{ED}=\{env_1, env_2, \ldots, env_m\}$ est l'ensemble fini des environnements de l'entrepôt,*
- *$Config^{ED}$ est l'ensemble des paramètres de configuration, visant à définir différents paramètres globaux.*

2.5. Exemple d'un entrepôt historisé

Nous présentons dans ce paragraphe un exemple d'entrepôt de données historisées. Nous précisons que les travaux initiés dans la thèse d'Olivier Teste (Teste, 2000) ont proposé un outil de conception du schéma de l'entrepôt à partir des sources opérationnelles. Nous utilisons cet outil pour la représentation du schéma de notre exemple d'entrepôt.

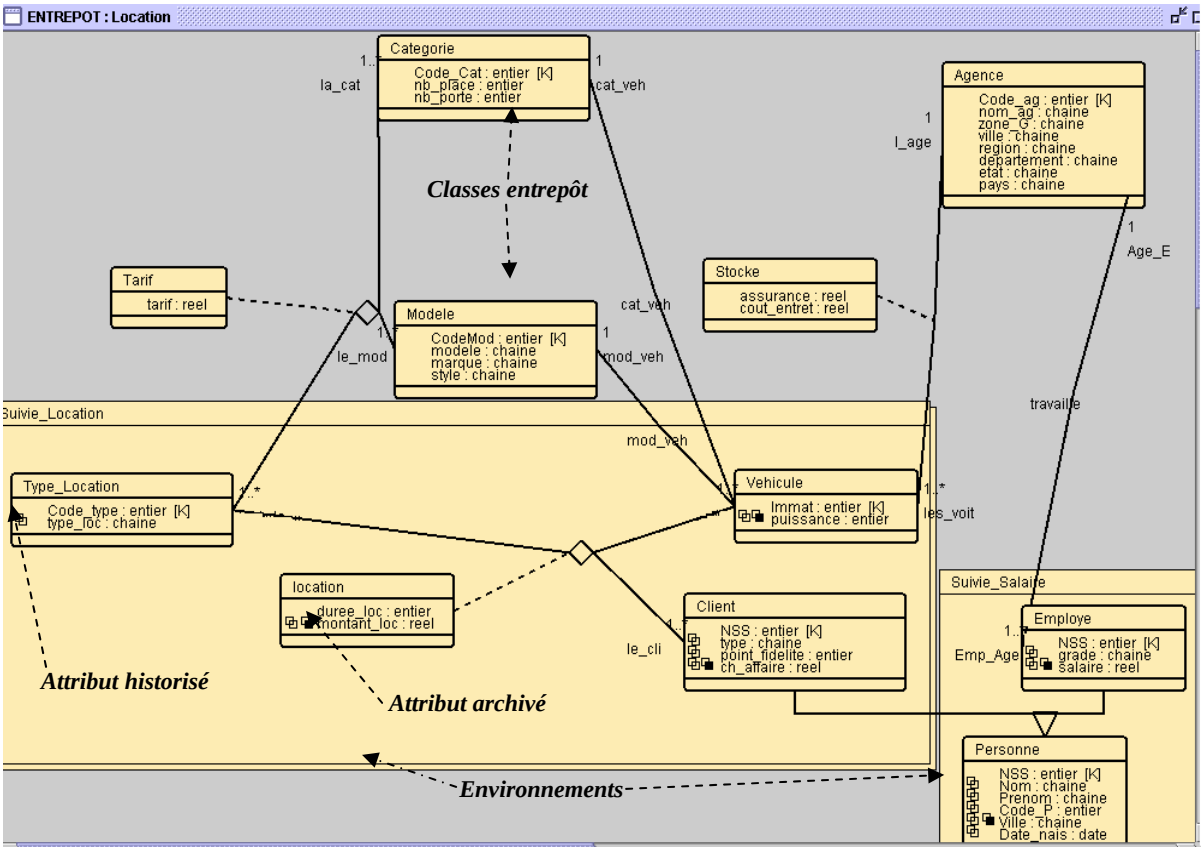


Figure IV.3 : *Schéma de l'entrepôt de données selon le diagramme de classes UML étendu*

Nous souhaitons modéliser une application de gestion de location de véhicules d'une société qui possède plusieurs agences en France et aux États-unis. Une location est réalisée selon plusieurs types (forfait journalier, heure, kilométrage, ...). Elle concerne un seul client et un seul véhicule. Un véhicule est caractérisé par sa catégorie et son modèle. Chaque type de location possède un tarif de location qui dépend de la catégorie et du modèle du véhicule.

Pour répondre à ces besoins, nous avons construit l'entrepôt Location présenté par le diagramme de classes UML de la Figure IV.3.

Nous souhaitons également conserver l'historique des locations journalières et réaliser le suivi mensuel des salaires des employés. Pour cela nous avons défini les environnements 'Suivi_Location' et 'Suivi_Salaire'. L'environnement 'Suivi_Location' comporte les classes *Client, Type_Location, Véhicule et Location*. La création de ces environnements permet de définir l'ensemble des informations à historiser par jour et par mois. L'historisation permet de garder le détail des évolutions des attributs dans le temps (les différentes valeurs affectées à chaque attribut dans le passé). L'ensemble des attributs historisés dans une classe forme son filtre temporel. L'archivage de ces attributs, leur affectation au filtre d'archives, permet de résumer une partie de ces valeurs pour ne conserver que l'information pertinente à l'utilisateur. Le deuxième environnement 'Suivi_Salaire' permet d'historiser les informations relatives aux salariés et notamment l'évolution de leur salaire et les changements de leur adresse dans le temps.

Dans la section suivante, nous proposons une méthode de conception de bases de données dimensionnelles à partir d'un entrepôt de données historisées.

3. Méthode de conception de base dimensionnelle

Notre objectif est de construire des bases de données dimensionnelles dans un contexte décisionnel ; nous intégrons à la fois la description des données sources (l'entrepôt) et la description des besoins des décideurs afin de tenir compte de toutes les données pertinentes à la prise de décision (Trujillo et al, 2003). Pour répondre à ce besoin, nous proposons une méthode de conception de bases dimensionnelles qui se décompose comme suit :

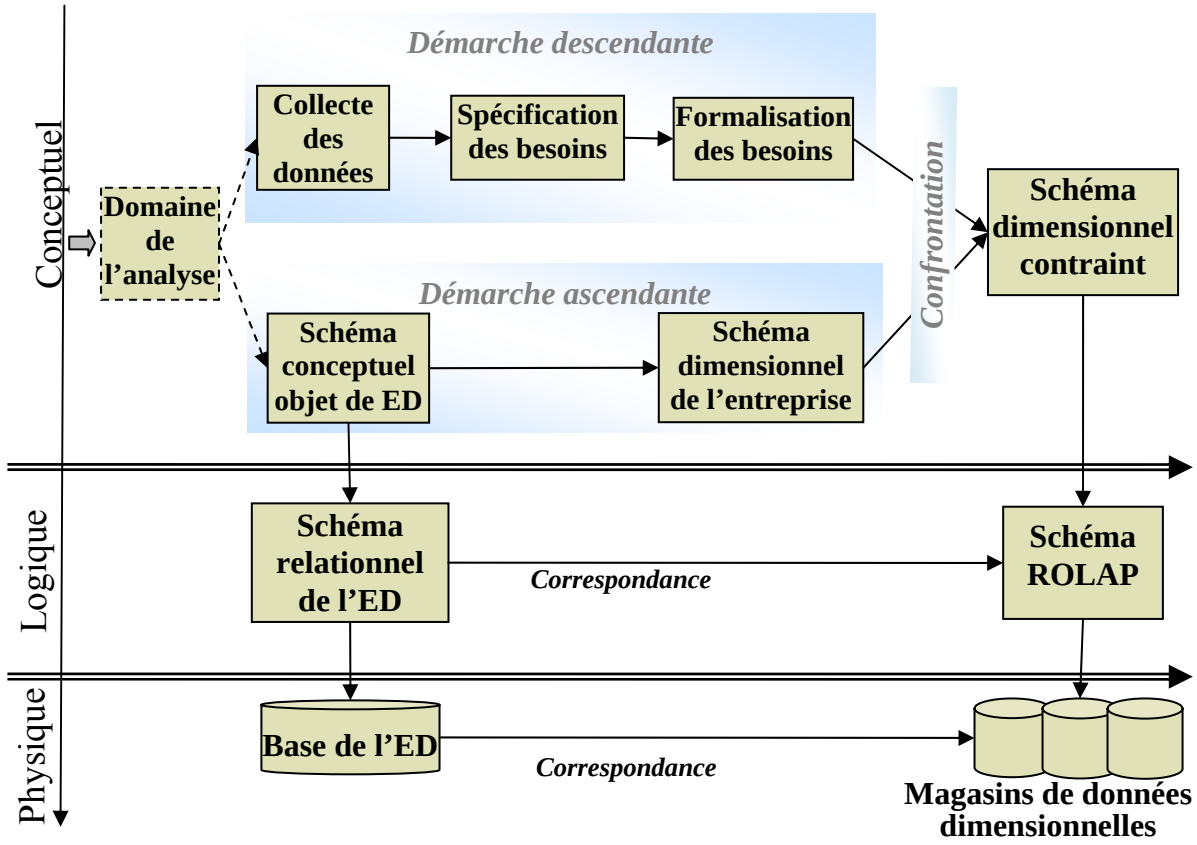


Figure IV.4 : *Etapes de notre méthode de conception de base dimensionnelle*

La méthode proposée permet de concevoir la base dimensionnelle selon les trois niveaux d'abstraction (conceptuel, logique et physique) :

- au niveau conceptuel, nous proposons de concevoir le schéma conceptuel des besoins décisionnels basé sur le formalisme de notre modèle dimensionnel contraint. Afin d'obtenir ce schéma dimensionnel, nous proposons une méthode mixte (Trujillo et al, 2003) comportant deux démarches complémentaires : une démarche descendante basée sur la spécification des besoins décideurs et une démarche ascendante basée sur le schéma des données de l'entrepôt. Les schémas dimensionnels résultants de ces deux démarches sont fusionnés dans une dernière étape de confrontation. Afin que les deux démarches, descendante et ascendante, convergent vers le même schéma dimensionnel, nous définissons en amont une étape qui définit le domaine d'analyse d'une manière générale ;
- au niveau logique, nous réalisons la transformation du schéma dimensionnel obtenu à partir de la phase précédente en un schéma relationnel OLAP (ROLAP). Ainsi, nous transformons les faits et les dimensions en tables relationnelles (Kimball et al, 2002) ;
- au niveau physique, nous implantons le schéma ROLAP dans une base de données ORACLE 9i.

Dans ce chapitre, nous présentons la phase conceptuelle de notre méthode de conception de bases de données dimensionnelles. Les phases logique et physique sont présentées dans le cadre de notre système d'aide à la conception de bases de données dimensionnelles présenté dans le chapitre V.

L'objectif de la phase conceptuelle est de modéliser les données décisionnelles d'une manière fiable, cohérente et complète répondant aux besoins des utilisateurs en terme d'aide à la prise de décision. Pour cela, nous nous sommes basés sur une approche mixte intégrant la description des besoins des décideurs à l'aide d'une démarche descendante et la description des données sources à l'aide d'une démarche ascendante. Le résultat de la phase conceptuelle est un schéma dimensionnel contraint obtenu après confrontation des schémas dimensionnels résultant des deux démarches.

Néanmoins, la mise au point de ces deux démarches nécessite la définition en amont du cadre général de l'application décisionnelle. En effet, nous avons été confrontés au problème de divergence des résultats des deux démarches dans un contexte de système décisionnel multidomaines.

Pour éviter cette divergence, nous avons proposé une étape de définition du ***domaine de l'analyse*** en amont des deux démarches descendante et ascendante. Cette étape permet aux concepteurs du schéma dimensionnel de partir sur une même base avec un domaine de conception bien défini. Lors de cette étape, nous proposons de fixer le champ de l'analyse en choisissant de travailler sur un métier ou une analyse donné de l'entreprise tels que le commercial, le marketing, la gestion des stocks, etc. La définition de ce cadre d'analyse permet au concepteur de choisir le groupe des décideurs à partir duquel il collectera les besoins décisionnels lors de la démarche descendante. Au niveau de la démarche ascendante, la définition du domaine d'analyse évite au concepteur de se perdre dans un schéma des sources qui couvre toutes les activités de l'entreprise.

Nous décrivons, dans les prochaines sections, nos démarches, descendante et ascendante, et l'étape de confrontation des résultats de ces démarches.

4. Démarche descendante

L'objectif de cette démarche est de concevoir un schéma dimensionnel en se basant sur les besoins des décideurs et sur les règles de gestion relatives aux données décisionnelles. Cette démarche se base sur trois étapes ; (1) la collecte des données, (2) la spécification des besoins et (3) la formalisation des besoins.

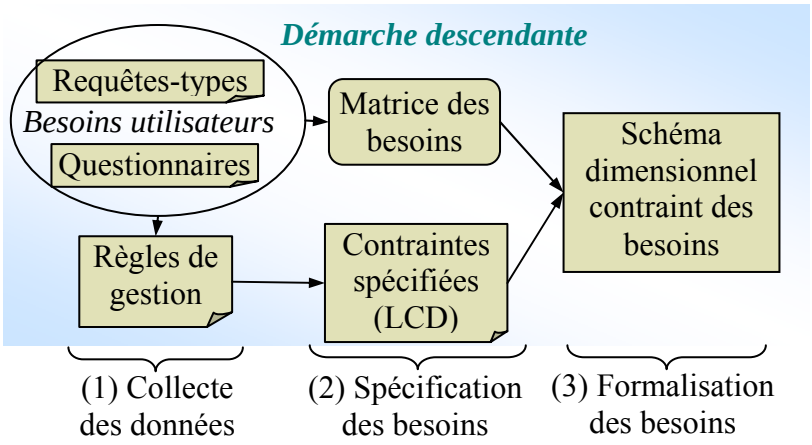


Figure IV.5 : *Démarche descendante*

La Figure IV.5 représente les différentes étapes de cette démarche. La première étape est consacrée à la collecte des données en se basant sur les questionnaires, les interviews des décideurs et les anciens rapports de l'organisation. Cette collecte permet d'exprimer les requêtes des décideurs sous un simple format et de rassembler les règles de gestion appliquées sur les données décisionnelles. Une deuxième étape permet de spécifier les besoins décisionnels à partir des données collectées. Dans cette étape, nous organisons les besoins dans une matrice afin d'extraire les éléments du schéma dimensionnel et nous définissons les contraintes appliquées sur ces besoins. Finalement, l'étape de formalisation permet de représenter les besoins à l'aide d'un schéma dimensionnel contraint.

Une plus ample description de ces étapes est présentée dans les paragraphes suivants.

4.1. Collecte des données

Dans cette étape, nous déterminons les besoins décisionnels initiaux en définissant les types d'informations susceptibles d'intéresser chaque groupe de décideurs. On procède alors à :

- la collecte des **requêtes-types** pertinentes en interviewant les décideurs,
- la mise au point d'un **questionnaire** permettant de mieux caractériser et identifier les besoins des décideurs,
- l'analyse des données décisionnelles et des résultats des interviews et des questionnaires afin de dégager les **règles de gestion** appliquées à ces données.

4.1.1. Requêtes-types

Les requêtes-types des décideurs peuvent être collectées à partir des anciens rapports d'analyse ou bien en les interrogeant. Nous proposons un pseudo-langage de structuration des requêtes qui permet de faciliter la définition des besoins. Ce pseudo-langage est basé sur trois clauses qui répondent à différents types de questions :

- la clause ***Analyser*** répond à la question ***Quoi ?*** Elle définit les données que les décideurs souhaitent analyser ;
- la clause ***En fonction*** répond aux questions ***Qui ? Où ?*** et ***Quand ?*** Elle indique les paramètres de l'analyse des données décrites par la clause ***Analyser*** ;
- la clause ***Pour*** répond à la question ***Pour Qui*** ou ***Quelles données*** ? Elle comporte des restrictions sur les données à analyser en fixant par exemple la valeur d'un paramètre d'analyse.

Exemple 1

Dans notre application de location de véhicules, l'analyse des rapports d'activité existants et une première interview des décideurs a permis de spécifier six requêtes-types (***R1*** à ***R6***). La Figure IV.6 présente un exemple de rapports d'activité que nous avons collecté pendant l'analyse. Le premier rapport décrit les montants des locations réalisées par mois et par agence. Le deuxième présente les montants des locations par ville. A partir de ces rapports, nous avons extrait les requêtes-types ***R2*** et ***R3***.

Listing des montants des locations par Mois et par Agence :

Mois Location	*Agence*	*Montant Location*
Janvier-00	Ag_tlse	90
Janvier-00	Ag_bord	120
Juin-00	Ag_tlse	70
juin-00	Ag_Dallas	110
Juin-00	Ag_Gren	200
Janvier-01	Ag_tlse	250
…	..	…

Listing des montants des locations par Ville :

Ville	*Montant Location*
Toulouse	410
Bordeaux	120
Grenoble	200
Dallas	110
…	
..	…

Figure IV.6 : *Un exemple de rapports d'activité*

R1 : ***Analyser*** le montant des locations

En fonction des mois et des véhicules

Pour les véhicules de type sport.

R2 : ***Analyser*** le montant des locations

En fonction des mois et des agences

R3 : ***Analyser*** le montant des locations

En fonction des villes

R4 : ***Analyser*** le chiffre d'affaire des employés

En fonction des mois

Pour l'employé Paul et l'année 2002.

R5 : ***Analyser*** les marques des véhicules

En fonction de la durée de location

Pour l'état de Floride.

R6 : ***Analyser*** les montants des locations

En fonction des véhicules

Pour la marque Peugeot.

4.1.2. Questionnaires

Le questionnaire est un outil de collecte de données dans lequel le décideur est guidé au travers d'un ensemble de questions dans le but d'obtenir des informations précises, nécessaires à la modélisation dimensionnelle. Ce questionnaire est établi en fonction des

résultats de la première étape, les requêtes, afin de compléter la description des besoins des décideurs. Ainsi, en remarquant que les véhicules sont organisés suivant une classification prédéfinie, le concepteur de la base dimensionnelle peut poser la question : comment sont classifiées les véhicules de location ?

Ce questionnaire doit permettre de modifier et éventuellement, d'ajouter de nouveaux éléments dans les requêtes-types :

- au niveau des axes de l'analyse, il permet de modifier ou d'ajouter de nouveaux axes, de vérifier s'il n'y a pas de nouvelles perspectives (hiérarchies). Au niveau de celles-ci, le questionnaire doit permettre d'organiser les niveaux de granularité et de compléter leur sémantique par des descripteurs (par exemple, ajouter un nom d'agence si son code n'est pas assez explicite pour le décideur) ;
- au niveau des indicateurs d'analyse, il permet d'ajouter des précisions sur le calcul des indicateurs, notamment, d'indiquer les fonctions d'agrégation compatibles et de déterminer le format des données (ex. données numériques ou textuelles). En outre, ce questionnaire permet de définir le niveau de détail des indicateurs, tels que le niveau journalier ou horaire des ventes. La définition du niveau de détail est primordiale dans la modélisation des besoins car un niveau de détail très bas engendre un volume de données très important et peut être non utilisable tandis qu'un niveau très haut engendre la perte des données détaillées ;
- au niveau général, il permet d'aider le décideur à spécifier ses besoins et les anticiper. En se basant sur les requêtes, le concepteur peut, avec son expérience dans les systèmes décisionnels, proposer de nouveaux indicateurs susceptibles d'intéresser le décideur. Par exemple, il peut lui proposer un ratio comparant le montant des ventes au nombre des clients de chaque magasin dans une application commerciale.

4.1.3. Règles de gestion

Les règles de gestion régissent le système d'information. Ces règles sont souvent représentées sous forme de contraintes qui doivent être respectées pour permettre le bon fonctionnement du système opérationnel.

Cette étape est réalisée en parallèle avec les deux étapes précédentes : les requêtes et les questionnaires.

Dans un premier temps, nous dégageons les règles de gestion relatives aux données décisionnelles. Ces règles sont appliquées dans le système opérationnel et peuvent être spécifiées dans la documentation et les rapports d'activité de ce système. Parmi les règles de gestion dans un système opérationnel, nous retrouvons l'organisation des produits dans des rayons par catégorie ou la classification des tarifs des véhicules par type et par niveaux de confort. L'interview des décideurs pour la définition des requêtes peut révéler d'autres règles de gestion.

Exemple 2

Dans notre exemple de société de location, nous avons spécifié une première règle relative à la classification des véhicules dans les agences de location. En effet, l'analyse de la

documentation de la société nous a permis de détecter que les agences françaises utilisent une classification de véhicule différente des agences américaines.

Dans un second temps, nous validons et complétons la liste de ces règles en proposant dans le questionnaire des questions sur les contraintes relatives au bon fonctionnement du système opérationnel et à l'organisation des entités de ce système.

Exemple 3

La validation des règles définies dans notre exemple nous a amené à ajouter une nouvelle règle relative à la localisation des agences françaises et américaines. En effet, les villes françaises sont organisées par région alors que les villes américaines sont organisées par état.

4.2. Spécification des besoins

En sortie de l'étape précédente, nous avons obtenu une liste de requêtes-types formulées à l'aide de notre pseudo langage et un ensemble de règles de gestion. L'étape de spécification permet d'analyser les données collectées afin de spécifier les besoins des décideurs. Ces besoins seront organisés en terme de paramètres et de mesures afin de préparer la définition du schéma dimensionnel.

A la fin de cette étape, nous obtiendrons une matrice des besoins reliant les mesures aux paramètres qui les analysent et un ensemble de contraintes non formalisées définies en se basant sur les règles de gestion.

4.2.1. Matrice des besoins

En se basant sur la formulation simplifiée des requêtes décrivant les besoins des décideurs, nous procédons à la définition de la matrice des besoins en trois étapes :

- une étape de **construction** de la matrice en fonction des propriétés de l'analyse ;
- une étape de **remplissage** de la matrice afin de caractériser les propriétés reliées par une relation d'analyse ;
- une étape de **simplification** de la matrice pour obtenir les mesures et les paramètres de l'analyse.

- *Construction de la matrice*

Durant l'étape de construction, nous définissons une liste de propriétés extraites des requêtes obtenues de l'étape de collecte de données. Nous construisons, ensuite, une matrice carrée dont les entêtes des lignes et des colonnes sont composées de la liste des propriétés (voir Tableau IV.1).

- *Remplissage de la matrice*

Durant l'étape de remplissage, nous répondons aux questions, quelles sont les données analysées et en fonction de quelles données. La structure de nos requêtes-types permet de répondre à ces questions :

- la clause ***Analyser*** permet d'indiquer les propriétés analysées ;

- les clauses ***En fonction*** et ***Pour,*** permettent d'extraire les propriétés qui paramètrent l'analyse.

Dans la matrice, chaque case cochée (√) indique que la propriété en ligne est analysée en fonction de celle en colonne.

Exemple 4

Considérons la première requête *R1* :

Analyser le montant des locations

En fonction des mois et des véhicules

Pour les véhicules de type sport.

A partir de cette requête, nous allons extraire tout d'abord les propriétés : le montant des locations, le mois représenté par le numéro de mois dans l'année, l'immatriculation identifiant un véhicule et le type de véhicule.

La propriété « montant des locations », extraite à partir de la clause ***Analyser,*** est analysée en fonction des propriétés mois, immatriculation et type extraites à partir des clauses ***En fonction*** et ***Pour***. Nous cochons les cases correspondantes à l'intersection entre la ligne de la propriété montant des locations et les colonnes des propriétés mois, immatriculation et type de véhicule.

Matrice des besoins

	Mt locations	***Ville***	***Région***	***Etat***	***Année***	***Mois***	***Nb jours***	***CA***	***Id_Emp***	***Id_Client***	***Immat***	***Type***	***Marque***
Mt locations		√	√	√	√	√				√	√	√	√
Ville													
Région													
Etat													
Année													
Mois													
Nb jours		√	√	√	√	√				√	√	√	√
CA					√	√			√				
Id_Emp													
Id_Client													
Immat													
Type													
Marque				√			√						

Tableau IV.1 : *Matrice carrée des propriétés de notre exemple*

Cette première matrice facilite l'automatisation de la définition des paramètres et des mesures de l'analyse. En effet, à partir de cette première matrice, nous pouvons dégager les propriétés qui ne décrivent aucune autre propriété. Celles-ci représentent les mesures ou les indicateurs d'activité (présentées en lignes). Les autres propriétés représentent les paramètres de l'analyse (présentées en colonnes).

- *Simplification de la matrice*

La simplification de la matrice des propriétés est réalisée en deux étapes :

- chaque colonne vide est supprimée de la matrice ; ceci permet d'enlever la propriété correspondante, de la liste des paramètres. En effet, une colonne vide indique que la propriété correspondante ne décrit aucun indicateur d'analyse et donc qu'elle ne fait pas partie des paramètres de l'analyse ;
- chaque ligne vide est supprimée de la matrice ; ceci permet d'enlever la propriété correspondante, de la liste des indicateurs. Cette propriété n'est analysée en fonction d'aucun paramètre (sa ligne est vide). Elle ne correspond pas à un indicateur dans notre analyse dimensionnelle.

Après cette simplification, nous pouvons retrouver des propriétés qui sont à la fois des indicateurs et des paramètres (exemple : Nb jours et Marque). Pour traiter ce cas particulier, nous procédons à une nouvelle lecture de la matrice permettant de montrer si ces propriétés sont plus utilisées en tant qu'indicateur ou en tant que paramètre. Nous notons que le type de la propriété peut déterminer sa nature. Souvent les indicateurs sont des propriétés numériques alors que les paramètres sont des descripteurs de type textuel.

Dans notre exemple, la propriété nombre de jours de location apparaît comme indicateur dans plusieurs colonnes et comme paramètre dans une seule ligne correspondant à la propriété *Marque (*voir la requête *R5)*. En outre, le nombre de jours des locations est de type numérique d'où le choix de la définition de cette propriété comme un indicateur. Cette simplification engendre la suppression de la seule case où la propriété *Nbjours* apparaît comme paramètre et par la suite la suppression de cette propriété de la liste des paramètres. De même, la suppression de la case où la propriété *Marque* apparaît comme indicateur implique la suppression de cette propriété de la liste des indicateurs. On obtient alors la matrice des besoins du Tableau IV.2. Cette simplification ne signifie pas que le besoin d'analyser cette propriété, exprimé par les utilisateurs, sera ignoré.

	Matrice des besoins									
Paramètres / Indicateur	***Ville***	***Région***	***Etat***	***Année***	***Mois***	***Id_Emp***	***Id_Client***	***Immat***	***Type***	***Marque***
Mt locations	√	√	√	√	√		√	√	√	√
Nb jours	√	√	√	√	√		√	√	√	√
CA				√	√	√				

Tableau IV.2 : *Matrice des besoins de notre exemple*

4.2.2. Contraintes spécifiées

Cette étape permet de spécifier les contraintes extraites à partir des règles de gestion définies dans l'étape de collecte des besoins. En effet, une règle de gestion peut donner lieu à plusieurs contraintes appliquées à différentes données dimensionnelles. Cette spécification permet de préparer l'étape suivante d'intégration des contraintes dans le schéma dimensionnel.

Afin d'exprimer les contraintes relatives au contexte dimensionnel, nous proposons un langage qui s'inspire du langage des contraintes objet OCL[1]. Nous appelons ce langage LCD

[1] http://www.omg.org/docs/formal/03-03-13.pdf

pour Langage de Contraintes Dimensionnelles. Ce langage permet de spécifier les contraintes sous un langage simple et facile à décrire par le concepteur.

Avant de décrire les concepts de notre langage, nous présentons le langage OCL dont nous nous sommes inspirés.

- *OCL (Langage de Contrainte Object)*

OCL est un langage formel standardisé par l'OMG[2] pour l'expression de contraintes dans un diagramme UML. Ces contraintes sont des informations supplémentaires qui ne peuvent pas être spécifiées directement avec le formalisme de base. La spécification de ce langage est basée sur les éléments suivants :

- les invariants regroupent les contraintes fixes dans un contexte donné. Le contexte d'une contrainte OCL, énoncé par le mot clé ***context***, peut être les classes ou les types UML. Une contrainte de type invariant est énoncée par le mot clé ***inv ;***
- les pré/post conditions s'appliquent sur les méthodes des classes objet. Les contraintes contenant le mot clé ***pre*** sont vérifiées avant l'exécution des méthodes et celles contenant ***post*** sont vérifiées après l'exécution des méthodes de classe.

Exemple 5

Pour exprimer le fait que toute personne doit être majeure, il est possible de définir une contrainte OCL exprimée sur la propriété *âge* de la classe personne de la manière suivante :

context p : Personne

inv : p.âge > 18

- *LCD (Langage de Contrainte Dimensionnel)*

LCD est un langage formel pour l'expression de contraintes dans un modèle dimensionnel. Ce langage s'inspire des concepts d'OCL. En effet, pour la spécification des contraintes, nous reprenons un concept de base du langage OCL à savoir le concept d'invariant. Dans notre contexte, les invariants sont définis sur les concepts dimensionnels, à savoir les faits, les dimensions et les hiérarchies. Ces concepts représentent dans notre langage le contexte d'une contrainte énoncé par le mot ***context***. Ainsi, nous pouvons définir, par exemple, la dimension *Agences* en tant que contexte d'une contrainte invariante énoncée par le mot clé ***inv***.

Exemple 6

Afin de spécifier les contraintes définies dans l'étape précédente, nous devons les analyser sémantiquement afin de pouvoir les exprimer dans notre langage *LCD*. Ces contraintes sont souvent basées sur les domaines des paramètres et des mesures de l'analyse. Une solution possible pour la détermination de ces contraintes est l'analyse des relations entre les domaines des paramètres et des mesures. Nous avons adopté cette solution pour la traduction en langage *LCD* des contraintes définies dans notre application de location de véhicules.

[2] http://www.omg.org/

a. L'organisation géographique française est différente de l'organisation américaine. Cette contrainte est appliquée sur les domaines des paramètres de l'axe d'analyse *Agences*. Une analyse de ces domaines permet de constater que les agences qui sont localisées dans un état américain (la valeur du paramètre *Etat* est non nulle) ont des régions nulles. Inversement, les agences françaises dont le paramètre *Région* est renseigné (différent de nul) ont un paramètre *Etat* vide.

***Context** a : Agences*
***Inv :** IF a.Etat -> notEmpty() then*
a.Région -> IsEmpty( )
Else
a.Région -> notEmpty( )
EndIF

b. Les agences françaises utilisent la nomenclature des véhicules française et les agences américaines appliquent la nomenclature américaine

***Context** l.Location inv*
***Inv :** IF l.Agences.Etat -> notEmpty() then*
l.Véhicules.Marque -> IsEmpty( )
Else
l.Véhicules.Type -> IsEmpty( )
EndIF

4.3. Formalisation des besoins

Après avoir collecté et spécifié les besoins des décideurs, nous réalisons dans cette étape la formalisation de ces besoins sous forme d'un schéma dimensionnel contraint. La conception de ce schéma est basée sur la matrice des besoins définie dans l'étape précédente. Les contraintes sémantiques définies dans l'étape de spécification sont intégrées à ce niveau dans le schéma dimensionnel des besoins.

4.3.1. Transformation de la matrice des besoins

A ce niveau, nous proposons de formaliser les besoins spécifiés dans les étapes précédentes sous forme de schéma dimensionnel. Nous considérons que ce schéma est le plus adapté à représenter les besoins des décideurs (cf. Chapitre I section I). En outre, cette représentation permet de faciliter l'étape de confrontation entre le résultat de la démarche descendante et celui de la démarche ascendante.

Pour obtenir ce schéma nous proposons les étapes suivantes qui se basent sur la matrice des besoins :

1) ***Définition des faits***. Durant cette étape nous regroupons les mesures dans des faits (sujets d'analyse). La définition des faits peut être réalisée d'une manière automatique en regroupant les mesures analysées au travers de paramètres identiques. Au niveau de cette étape, nous définissons aussi les fonctions d'agrégation compatibles avec chaque mesure en se basant sur les questionnaires.

Exemple 7

Pour notre exemple, nous regroupons les mesures « montant » et « nombre de jours » des locations dans le fait *Location* et nous définissons le fait *Performance* pour la mesure

« CA ». Pour chacune de ces mesures, nous définissons les fonctions d'agrégation nécessaires à l'analyse.

- *Location* ((Montant_Loc, {sum, avg}), (Nb_jours, {sum, avg}))
- *Performance* ((CA, {sum, avg, max, min}))

Une optimisation du résultat de cette opération est possible afin de prendre en compte la sémantique des mesures. Nous pouvons, par exemple, diviser un fait en deux faits en considérant que les mesures représentent des sémantiques différentes. Dans l'exemple d'une analyse commerciale, les indicateurs montants des ventes et stock de produit partageant les mêmes dimensions d'analyse (produit, magasin et temps), sont définies dans un même fait, avant cette optimisation. Ce fait est éclaté, par la suite, en considérant que l'analyse des ventes et le suivi des stocks représentent deux sujets différents d'analyse.

2) ***Définition des dimensions***. Cette étape réalise le ***regroupement*** des paramètres dans des dimensions (axes d'analyse), l'***enrichissement*** de ces dimensions à travers de nouvelles propriétés et la définition de la ***granularité*** de l'analyse.

a. *Le regroupement des paramètres*. De même que pour les faits, une automatisation de cette opération est possible en rassemblant les paramètres qui caractérisent les mêmes mesures. Les dimensions résultantes de cette automatisation sont traitées, par la suite, par le concepteur afin de dégager les groupes de paramètres qui caractérisent la même dimension d'un point de vue sémantique. Ce traitement sémantique se base, aussi, sur les questionnaires qui fournissent des informations sur les dimensions et les propriétés qui les caractérisent. Ainsi, c'est le concepteur qui définit les noms des dimensions en fonction de leur sémantique.

b. *L'enrichissement des dimensions*. Cette opération est basée sur les questionnaires au niveau desquels nous avons ajouté de nouveaux attributs (paramètres et attributs faibles) aux axes de l'analyse. Ces attributs sont greffés aux différentes dimensions de l'analyse.

c. *Définition de la granularité de l'analyse*. Nous définissons à cette étape les granularités de l'analyse ; nous déterminons le niveau d'analyse le plus fin pour chaque dimension. L'importance de la définition de la granularité est qu'elle fixe le niveau de détail des données analysées. Ainsi, nous ne pouvons plus retrouver les données d'un niveau de détail plus fin.

Exemple 8

Dans notre exemple, le regroupement automatique des paramètres engendre les dimensions suivantes :

- *D1* (Région, Etat, Id_Client, Immat Véhicule, Type, Marque),
- *D2* (Id_Employé),
- *D3* (Mois, Année).

L'analyse de ces dimensions permet d'éclater la dimension D1 en trois dimensions *Agences*, *Clients* et *Véhicules*. En effet, au niveau du questionnaire, le concepteur demande au décideur la nature et le type de chaque propriété et l'entité qu'elle caractérise. L'affectation des noms aux dimensions est réalisée par le concepteur en fonction de ces informations. Ainsi, nous obtiendrons les dimensions suivantes :

- *Agences* (Ville, Région, Etat),
- *Clients* (Id_Client),
- *Véhicules* (Immat véhicule, Type, Marque),
- *Employés* (Id_Employé),
- *Temps* (Mois, Année).

Au niveau de l'étape de la collecte des données, nous avons enrichi les dimensions de l'analyse par les informations collectées dans les questionnaires. Ainsi, un client est caractérisé par son nom, son prénom et sa ville. Les agences sont identifiées par un code (*Code_Ag*) et caractérisées par leur nom et par leur localisation (V*ille, Région, Etat, Pays*). Les employés sont caractérisés par leur nom, leur prénom, leur âge et la tranche d'âge à laquelle ils appartiennent.

L'étude de la granularité de l'analyse nous a permis d'ajouter d'autres paramètres aux dimensions. Ainsi, l'analyse des locations est réalisées par agence (*Code_Ag*), jour (*IdT*), client (*Id_Client*) et véhicule (*Immat*). L'analyse des chiffres d'affaire des employés est effectuée par employé (*Id_employé*) et par jour (*IdT*). Ces paramètres représentent le niveau de granularité le plus fin de l'analyse. Les dimensions obtenues après cet enrichissement sont les suivantes :

- *Agences* (Code_Ag, Nom, Ville, Département, Région, Etat, Pays),
- *Clients* (Id_Client, Nom, Prénom, Ville),
- *Véhicules* (Immat, véhicule, Type, Marque),
- *Employés* (Id_Employé, Nom, Prénom, Tranche),
- *Temps* (IdT, jour, Mois, Année).

3) ***Définition des hiérarchies***. Cette étape consiste à organiser les paramètres de chaque dimension dans des hiérarchies. Cette opération est réalisée en se basant sur le questionnaire et les règles de gestion définies dans l'étape de collecte de données. Pour la dimension *Agences*, par exemple, nous définissons deux hiérarchies : une hiérarchie relative à la localisation des agences françaises et une deuxième relative à la localisation des agences américaines.

4) ***Définition du schéma dimensionnel***. Finalement, nous réalisons l'affectation des dimensions aux faits. L'affectation est réalisée d'une manière automatique en se basant sur la matrice des besoins. Chaque dimension comportant des paramètres qui ont des intersections avec un indicateur d'un fait, est affectée à ce fait.

Exemple 9

Dans notre exemple, les locations sont analysées en fonction des dimensions *Agences, Temps, Clients* et *Véhicules.* Le fait Performance est analysé en fonction des dimensions *Employés* et *Temps.*

- *Location* : Agences, Temps, Clients, Véhicules.
- *Performance* : Employés, Temps

Le résultat de cette étape de formalisation est notre schéma en constellation qui représente les besoins des décideurs (Figure IV.7).

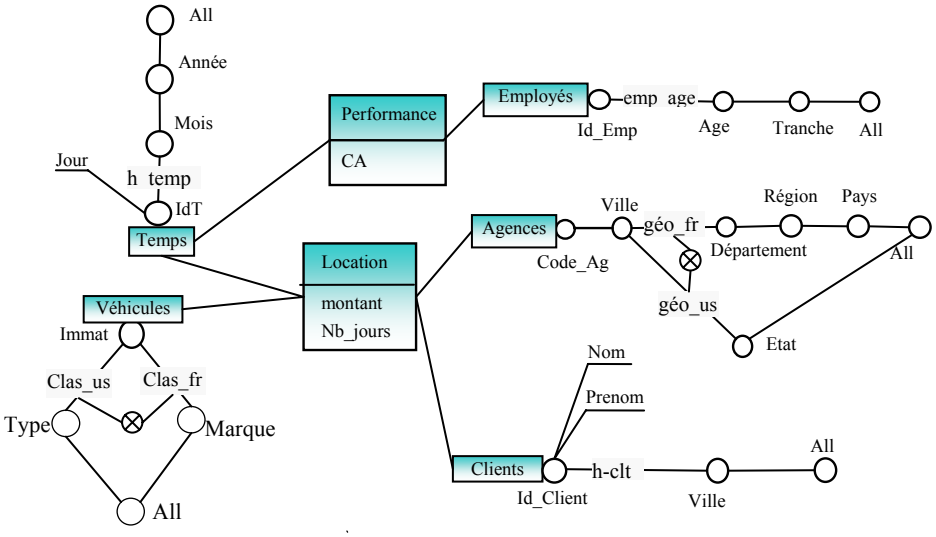


Figure IV.7 : *Schéma des besoins*

4.3.2. Intégration des contraintes

Dans cette étape nous réalisons l'intégration des contraintes définies dans l'étape de spécification dans le schéma des besoins. Ainsi, nous transformons les contraintes exprimées avec notre langage LCD en contraintes intra et inter-dimensions.

Exemple 10

La conception du schéma dimensionnel avec les différentes dimensions et hiérarchies, nous permet de définir les contraintes sémantiques.

La première contrainte définie dans notre exemple 5 relative à l'organisation géographique française et à l'organisation américaine, est exprimée par une contrainte d'exclusion intra-dimension entre les hiérarchies "*geo_fr*" et "*geo_us*" de la dimension *Agences*.

La deuxième contrainte caractérisant le fait que les agences françaises utilisent la nomenclature des véhicules française et les agences américaines appliquent la nomenclature américaine, est exprimé par une contrainte d'exclusion intra-dimension entre les hiérarchies "*clas_fr*"' et "*clas_us*" de la dimension *Véhicules*.

4.4. Bilan de la démarche descendante

La démarche descendante réalise progressivement la transformation des besoins des décideurs exprimés dans un langage naturel en un schéma dimensionnel structuré intégrant l'expression des contraintes. Cette transformation progressive permet au concepteur d'exprimer les besoins décisionnels de façon exacte, complète et fiable.

Durant cette démarche, nous proposons, tout d'abord, d'expliciter les besoins décisionnels en se basant sur des outils de collecte de données (cf. Figure IV.5) :

- les interviews permettant de définir un ensemble de ***requêtes-types*** structurées décrivant les besoins décisionnels ;

- les questionnaires complétant la sémantique des requêtes en enrichissant les informations collectées ;
- les règles de gestion relatives aux données décisionnelles.

Les requêtes serviront comme base à la construction de la matrice des besoins reliant les mesures aux paramètres de l'analyse. La définition des mesures et des paramètres dans la matrice simplifie et automatise le processus de création des faits et des dimensions du schéma dimensionnel. En parallèle, l'ensemble des règles de gestion servira de base pour la spécification des contraintes à l'aide de notre langage de contraintes dimensionnelles.

Cette démarche descendante constitue une première étape dans la conception du schéma dimensionnel au niveau conceptuel. Basée uniquement sur les besoins des décideurs, elle ne tient pas compte des données sources stockées dans l'entrepôt (Figure IV.1). Nous proposons dans la section suivante, une démarche ascendante qui répond à ce besoin en construisant un schéma dimensionnel à partir du schéma de l'entrepôt de données.

5. Démarche ascendante

Afin de tenir compte des informations contenues dans la source de données (l'entrepôt pour notre cas), nous proposons une démarche ascendante incrémentale. Cette démarche part du schéma conceptuel de l'entrepôt de données historisées pour construire le schéma dimensionnel contraint d'un magasin de données (Bret et al, 1999). Dans cette démarche, nous supposons que le concepteur du schéma dimensionnel a une double compétence informatique et métier. En effet, c'est le concepteur du schéma dimensionnel qui doit détecter les différents centres d'intérêt de l'organisation en analysant le schéma de l'entrepôt. En outre, l'étape de la définition du domaine d'analyse permet de fournir au concepteur une vision générale des besoins décisionnels de l'entreprise. Dans notre méthode, nous considérons que la démarche ascendante est réalisée en parallèle avec la démarche descendante. Toutefois, le concepteur peut, au niveau de la phase ascendante, tenir compte des informations collectées lors de la démarche descendante.

L'objectif de cette démarche est de concevoir un schéma dimensionnel en se basant sur le schéma conceptuel de l'entrepôt de données et sur le domaine de l'analyse. Le schéma obtenu à partir des données de l'entreprise stockées dans l'entrepôt de données est appelé schéma dimensionnel de l'entreprise afin de le différencier du schéma des besoins qui part des données des décideurs.

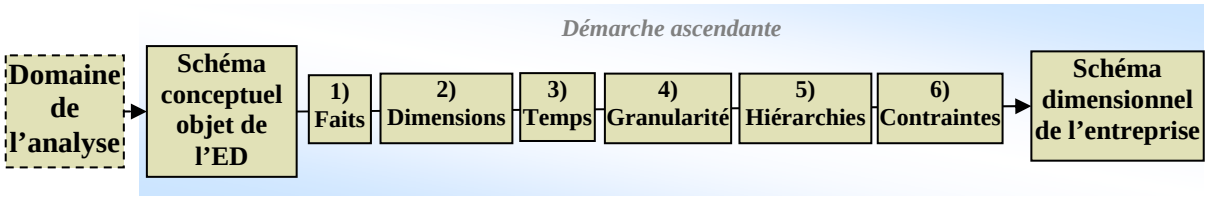


Figure IV.8 : *Etapes de la démarche ascendante.*

Notre démarche ascendante comporte six étapes :

1) Détermination des faits représentant les sujets analysés.
2) Détermination des dimensions représentant les perspectives de l'analyse.

3) Définition de la dimension temporelle et détermination des faits qui seront reliés aux hiérarchies temporelles détaillées et/ou d'archives.
4) Définition des granularités des données de l'analyse.
5) Organisation des paramètres des dimensions selon des dépendances hiérarchiques pour supporter les analyses à différents niveaux de détail.
6) Expression des contraintes sémantiques inter et intra dimension.

Les différentes étapes sont illustrées par des exemples basés sur l'entrepôt de données présenté dans le paragraphe 2.4.

5.1. Détermination des faits

La détermination des sujets de l'analyse permet de dresser la liste des faits du modèle dimensionnel. Selon notre démarche, un fait est projeté à partir d'une classe représentative de l'entrepôt. Cette classe est choisie par le concepteur de la base dimensionnelle. Celui-ci analyse la source (dans notre cas c'est l'entrepôt) et définit les classes susceptibles d'être transformées en sujets d'analyse. Une classe représentative est définie comme suit :

Définition

*Une **Classe Représentative** (CR) enregistre les détails d'événements particuliers comme les salaires, les réservations d'hôtels. Ce sont ces événements, entre autres, que les décideurs veulent analyser. Une classe représentative :*

- *décrit un événement qui se produit à un instant donné ;*
- *contient les mesures, comme le montant en devise, le poids, les volumes. Ces mesures forment la base des résultats que l'entrepôt permet d'étudier.*

Souvent, le concepteur choisit comme classes représentatives les classes qui comportent des propriétés numériques transformables en indicateurs d'analyse. Ces propriétés sont transformées en mesures à l'aide d'une fonction de calcul. Les mesures du fait sont donc obtenues en appliquant une fonction de calcul sur un ou plusieurs attributs de la classe représentative. Cette fonction peut être une simple fonction de projection (fonction identité) ou une expression mathématique (montant = prix unitaire * quantité).

Afin de permettre l'agrégation de ce fait, nous définissons pour chaque mesure l'ensemble des fonctions d'agrégation sémantiquement compatibles. Ces fonctions seront utilisées lors de la manipulation des données dimensionnelles en agrégeant les mesures à différents niveaux hiérarchiques.

Exemple 11

En se basant sur le schéma de l'entrepôt (cf. Figure IV.3) et sur le domaine d'analyse commercial de l'activité des agences de locations, le concepteur souhaite dégager les sujets d'analyse susceptibles d'intéresser les décideurs. Dans le domaine commercial, l'analyse du schéma permet de détecter une classe représentative qui comporte des indicateurs d'analyse ; la classe *Location* comportant les montants des locations des agences. Nous souhaitons, maintenant, définir un fait analysant les montants et les durées des locations. Ce fait répond aux besoins décisionnels du domaine d'analyse défini. Pour répondre à ce besoin, nous définissons le fait '*Location_Veh*' à partir de la classe représentative

Location. Ce fait est défini par le couple (N^F, M^F) représentant le nom du fait et les mesures accompagnées de leurs fonctions d'agrégation.

- $N^{Location\_Veh}$ = "Location_Veh",
- $M^{Location\_Veh}$ = {(montant = ED.Location.Montant_Loc, {sum, avg}), (nbjours = ED.Location.durée_Loc, {sum, avg})} avec ED le nom de l'entrepôt de notre exemple.

5.2. Détermination des dimensions

Les dimensions sont élaborées à partir des classes entrepôts reliées à la classe représentative (autres que les dimensions temporelles) (Bret et al, 1999), appelées classes ***déterminantes***. La détermination de ces classes est réalisée automatiquement en suivant le principe de dépendance fonctionnelle entre classes.

Définition

*Une classe Ci est **déterminante** d'une classe CR noté Ci => Cr si :*

- *Ci = Cr,*
- *Ci hérite de Cr,*
- *Ci représente une classe d'association et Cr entre dans la formation de l'association.*
- *Ci est reliée à Cr par une relation d'association monovaluée (X, 1) ou*
- *Ci est reliée à Cr par une relation de composition de type (1, x) si Ci est composée de Cr ou (x, 1) si Ci compose Cr.*

La contrainte sur les cardinalités des relations d'association et de composition permet de garantir l'unicité entre les valeurs d'une classe C_1 et les valeurs liées d'une classe C_2. Cette propriété est essentielle, car elle permet de relier dans le magasin les mesures issues de la classe représentative aux paramètres issus des classes déterminantes.

Le principe de dépendance respecte la ***propriété de transitivité***. Cette propriété permet de définir une classe C_1 déterminante d'une classe C_3 si la classe C_1 est déterminante d'une classe C_2 elle-même déterminante de C_3 ; soient $C_1 \in C^{ED}$, $C_2 \in C^{ED}$ et $C_3 \in C^{ED}$, si $C_1 \Rightarrow C_2$ et $C_2 \Rightarrow C_3$ alors $C_1 \Rightarrow C_3$.

A partir du principe de dépendance et de sa propriété de transitivité, on peut déterminer l'ensemble ***Determin (CR) = {CD_1, CD_2,…, CD_m}*** des classes déterminantes d'une classe représentative *CR*. Leur intérêt est de déterminer l'ensemble des classes entrepôt à partir desquelles peuvent être créées les dimensions.

Exemple 12

L'application du principe de dépendance sur l'exemple de notre entrepôt de données nous amène à déterminer l'ensemble des classes déterminantes de la classe représentative *Location* :

Détermin (Location) = {*Location, Véhicule, Agence, Client, Personne, Type_Location, Catégorie, Modèle*}

Les dimensions du fait *Location_Veh* sont définies à partir de ces classes déterminantes. En se basant sur le domaine de l'analyse, le concepteur définit toutes les dimensions susceptibles d'intéresser les décideurs. Dans le domaine commercial, le concepteur choisit

de définir les dimensions *Agences*, *Véhicules*, *Clients* respectivement à partir des classes de l'entrepôt *Agence*, *Véhicule* et *Client*. Ces dimensions sont définies par les couples *(N^D, A^D)* représentant le nom de la dimension et ses attributs. Par exemple, la définition de la dimension *Agences* ($N^{Agences}$, $A^{Agences}$) est réalisée comme suit :

- $N^{Agences}$ = "*Agences*",
- $A^{Agences}$ = {Code_Ag = ED.Agence.Code_ag, Ville = ED.Agence.ville, Departement = ED.Agence.Departement, Region = ED.Agence.Region, Pays = ED.Agence.Pays, Etat = ED.Agence.Etat}.

5.3. Définition de la dimension temporelle

La définition de la dimension temporelle dépend de la configuration des environnements de l'entrepôt de données, des filtres temporels et des filtres d'archives de la classe représentative et des classes déterminantes. En effet, la structure spécifique des données temporelles, conservées sous forme détaillée et/ou archivée au niveau de l'entrepôt, rend leur exploitation plus complexe. Néanmoins, elle fournit des analyses plus précises pour l'aide à la décision.

Une hiérarchie temporelle détaillée est connectée à un fait F si et seulement si :

- la classe représentative CR du fait appartient à un environnement, et
- les mesures du fait sont issues d'attributs appartenant au filtre temporel de la classe représentative.

Une hiérarchie temporelle archivée est connectée à un fait F si et seulement si :

- la classe représentative CR du fait appartient à un environnement, et
- les mesures du fait sont issues d'attributs appartenant au filtre d'archives de la classe représentative.

Ces conditions vérifient que les paramètres temporels de détail (respectivement d'archives) ne caractérisent que les mesures qui possèdent un historique détaillé (respectivement archivé). Par exemple, dans notre application de location de véhicules, la mesure *montant* peut être analysée d'une manière journalière suivant la hiérarchie temporelle détaillée durant les années 1996 à 2004 puisque la valeur de l'attribut *Montant_loc* à partir duquel la mesure est calculée est conservée tous les jours. Avant cette période, cet attribut est archivé tous les trimestres, les valeurs de la mesure correspondant à cette date sont analysées en fonction de la hiérarchie d'archive.

Les données historisées sont analysées selon une hiérarchie compatible avec leur granularité de rafraîchissement au niveau de l'entrepôt. Par exemple, si on rafraîchit tous les mois, on ne peut pas dimensionner les données avec le paramètre jour, par contre les paramètres mois, trimestre et année sont compatibles.

L'analyse des données suivant les hiérarchies temporelles nécessite l'étude des cas où les autres dimensions de l'analyse proviennent de classes non historisées ou non archivées. Nous présentons dans les sections suivantes les deux cas possibles.

- *Cas où la classe déterminante n'est pas historisée*

Dans ce cas, l'analyse des mesures provenant de la classe représentative historisée est réalisée en fonction des valeurs de paramètres non historisées. L'évolution des valeurs de ces derniers n'a pas été jugée pertinente lors de la construction de l'entrepôt de données.

Exemple 13

Considérons l'exemple du fait *Location_Veh* où la dimension *Agences* est dérivée à partir de la classe déterminante *Agences* non historisée. L'analyse des montants des locations mensuelles par agence ne tient pas compte du changement du nom de l'agence codée « AG-FF4 ». Cette information jugée non pertinente n'a pas été historisée dans l'entrepôt.

- *Cas où toutes les classes (CR et CD) sont historisées et appartiennent au même environnement*

Dans ce cas, chaque valeur de mesure est analysée en fonction des valeurs des paramètres qui correspondent à la même période d'historisation. Par exemple, si un client change de ville de résidence en l'année '2000' et déménage de la ville de 'Toulouse' vers 'Bordeaux', alors l'analyse des locations en fonction des villes des clients doit affecter les montants des locations de ce client pour la période antérieure à '2000', à la ville de 'Toulouse' et les montants des ventes postérieures à cette date, à la ville de 'Bordeaux'.

Après avoir défini les dimensions et les attributs qui les composent, le concepteur doit définir la granularité de l'analyse des faits.

5.4. Définition de la granularité de l'analyse

La quatrième étape consiste à spécifier quel est le niveau le plus détaillé suivant lequel les données dimensionnelles sont analysées. La définition des différentes dimensions connectées à un fait ne détermine pas le niveau de granularité de l'analyse. En effet, les mesures du fait peuvent être observées à différents niveaux de détails. Par exemple, pour une dimension géographique, les mesures peuvent être définies au niveau *Département* ou bien *Ville*. Au niveau de la démarche ascendante, nous choisissons la granularité d'analyse la plus détaillée au niveau de chaque dimension en se basant sur le schéma de l'entrepôt.

La détermination de la granularité de l'analyse permet de définir les paramètres les plus détaillés de chaque dimension. Chacun de ces paramètres représente le début d'une structure hiérarchique que nous allons définir dans l'étape suivante.

5.5. Hiérarchisation des dimensions

Les paramètres des dimensions sont organisés en une structure hiérarchique qui permet d'analyser les mesures à différents niveaux de détail. Ainsi, nous définissons dans chaque dimension plusieurs hiérarchies de paramètres. La définition d'une hiérarchie est réalisée par la détection des **dépendances hiérarchiques** entre les paramètres d'une dimension (Bret et al, 1999).

Définition

Un paramètre p_i ***dépend hiérarchiquement*** *d'un autre paramètre p_j, noté pi ➔ p_j ssi :*

- *p_i dépend fonctionnellement de p_j, noté $p_i \rightarrow p_j$. Une dépendance fonctionnelle entre deux paramètres p_i et p_j, indique que chaque valeur de pj détermine d'une manière unique la valeur de p_i.*
- *p_j ne dépend pas fonctionnellement de p_i, noté $p_j \nrightarrow p_i$.*

En se basant sur ce principe, nous pouvons définir différentes hiérarchies dans la même dimension.

Au niveau de cette démarche, nous proposons de définir les hiérarchies les plus complètes que possible en fonction du schéma de l'entrepôt.

Exemple 14

L'analyse des dépendances fonctionnelles entre les paramètres de la dimension *Agences* nous a permis de définir les trois hiérarchies suivantes ;

- *geo_fr* = ('géo. française', Code_Ag, Ville, Département, Région, Pays, All),
- *geo_us*= ('géo. américaine', Code_Ag, Ville, Etat, Pays, All),
- *géo_zn* = ('zone .agence', Code_Ag, Zone, All).

5.6. Expression des contraintes

Les contraintes sont des expressions qui précisent le rôle ou la portée d'un élément de modélisation (elles permettent d'étendre ou de préciser sa sémantique) (Doucet et al, 1996). Ainsi, nous proposons un ensemble de contraintes associées aux concepts et aux données de notre modèle afin de donner l'interprétation la plus précise possible de la réalité pour une meilleure prise de décision. Nous définissons des contraintes à deux niveaux :

- les contraintes intra-dimensions sont des contraintes concernant une seule dimension, autrement dit, il s'agit de contraintes entre les hiérarchies d'une même dimension ;
- les contraintes inter-dimensions sont des contraintes portant sur plusieurs dimensions.

5.6.1. Contraintes intra-dimension

Nous rappelons les différentes contraintes qui peuvent êtres définies au niveau d'une dimension : *exclusion*, *inclusion, partition, simultanéité* ou *totalité*. La détection de ces contraintes est basée sur l'analyse des données de l'entrepôt. Cette analyse au niveau de la classe *Agence*, nous a permis de constater que, pour toutes les agences françaises, l'attribut *Etat* est nul et que, pour toutes les agences américaines, les attributs *Département* et *Région* sont nuls. Ce fait est exprimé au niveau de notre schéma conceptuel à l'aide de *la condition* d'*appartenance* aux hiérarchies. Dans un second temps, nous constatons que toutes les agences de la dimension font partie des agences françaises ou bien des agences américaines.

Exemple 15

En se basant sur l'analyse des données de la classe *Agence* de l'entrepôt, nous avons défini les contraintes d'exclusion entre "*geo_fr*"et "*geo_us*" pour exprimer le fait que les agences

organisées suivant l'une des deux hiérarchies ne peuvent pas appartenir à la deuxième hiérarchie.

Par contre, nous remarquons que les agences situées dans les différents états (suivant la hiérarchie "*geo_us*") peuvent être organisées par *Zone* selon la hiérarchie "*geo_zn*". De même pour les agences décrites par la hiérarchie "*geo_fr*". Ainsi, nous avons défini une contrainte d'inclusion de la hiérarchie "*geo_us*" dans "*geo_zn*" et de même pour "*geo_fr*".

Les hiérarchies "*geo_fr*" et "*geo_us*" de la dimension Agence contiennent la totalité des agences de la dimension. Ceci est exprimé par une contrainte de totalité entre les deux hiérarchies.

Ces contraintes sont définies comme suit :

C1 :: ***geo_fr ⊗ geo_us*** C3 :: ***geo_fr ⊙ geo_zn.***

C2 :: ***geo_fr ⊖ geo_us*** C4 :: ***geo_us ⊙ geo_zn.***

5.6.2. Contraintes inter-dimensions

Il s'agit de contraintes portant sur les hiérarchies de dimensions distinctes reliées à un même fait. Nous rappelons que, comme au niveau intra, il existe cinq contraintes inter dimensions : *exclusion*, *inclusion**,** partition**,** simultanéité* ou *totalité**.*** De la même manière que pour les contraintes intra, les contraintes inter-dimensions sont définies suite à l'analyse des instances des classes représentatives et des classes déterminantes de l'entrepôt. Par exemple, l'analyse des données de la classe représentative *Location* nous permet de constater que les instances de cette classe, reliées aux instances des agences françaises, ne sont pas associées aux instances de la classe véhicule organisées suivant une classification américaine.

Exemple 16

En se basant sur l'analyse des données de l'entrepôt, une contrainte d'exclusion peut être définie entre la hiérarchie "*geo_fr*" de la dimension *Agence*s et la hiérarchie "*clas_us*" de la dimension *Véhicule*s. La première décrit l'organisation géographique française et la deuxième définit la classification des véhicules aux Etats-Unis. La contrainte d'exclusion décrit le fait que les mesures du fait ne peuvent pas être paramétrées par les deux hiérarchies à la fois. De même, une contrainte de totalité est définie entre ces deux hiérarchies exprimant le fait qu'à chaque instance du fait *Loc_Vehicule* correspond au moins une instance de l'une ou de l'autre des deux hiérarchies.

La définition d'une contrainte d'inclusion de la hiérarchie "*clas_us*" dans la hiérarchie "*geo_zn*" exprime le fait que l'ensemble des instances du fait *Loc_Vehicule* décrites selon la nomenclature américaine des véhicules ("*clas_us*") sont incluses dans les instances du fait associées à la hiérarchie "*geo_zn*" de la dimension *Agences*.

C5 :: ***geo_fr ⊗ clas_us*** C7 :: ***clas_us ⊙ géo_zn***

C6 :: ***géo_us ⊖ clas_fr*** C8 :: ***clas_fr ⊙ géo_zn***

5.7. Bilan de la démarche ascendante

Afin de tenir compte des données de l'entrepôt lors de la conception du schéma dimensionnel contraint des données décisionnelles, nous avons proposé une démarche ascendante. Cette démarche est composée de six étapes permettant au concepteur de concevoir progressivement le schéma dimensionnel. Une première étape consiste à détecter les classes de l'entrepôt représentatives d'un sujet d'analyse. A partir de chaque classe représentative, nous définissons automatiquement les classes déterminantes candidates pour devenir des dimensions d'analyse. Une fois que les faits et les dimensions sont définis, nous procédons à la définition de la granularité de l'analyse en choisissant les granularités les plus détaillées au niveau de l'entrepôt. Une cinquième étape réalise l'organisation des paramètres en hiérarchies multiples dans les dimensions en se basant sur le principe de dépendance. Et enfin, nous intégrons la définition des contraintes sémantiques dans notre schéma dimensionnel. Le résultat de cette démarche est un schéma dimensionnel intégrant l'expression des contraintes sémantiques (voir Figure IV.9). Ce schéma représente les données extraites de l'entrepôt de l'entreprise comportant toutes les données qui caractérisent son activité. Aussi, il englobe toutes les données détaillées que le décideur peut ne pas exprimer dans ses besoins (au niveau de la démarche descendante).

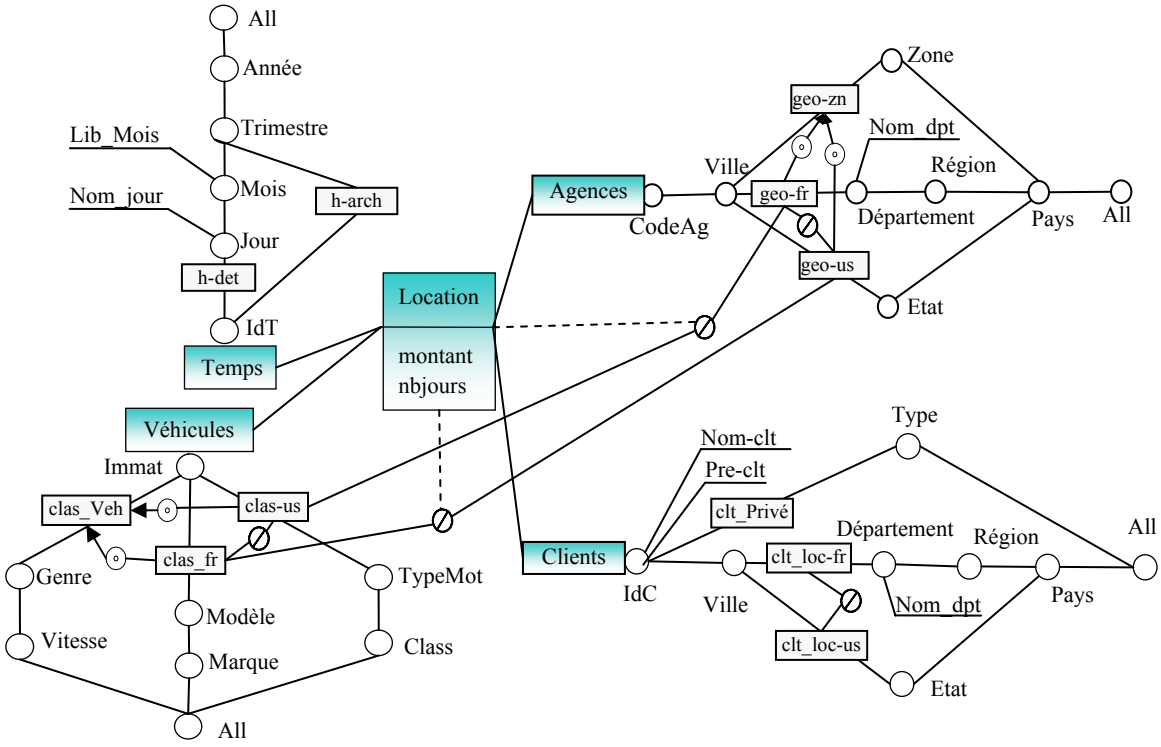


Figure IV.9 : *Schéma dimensionnel de l'analyse des locations selon la démarche ascendante.*

Pour valider les résultats des deux démarches descendante et ascendante, nous réalisons une confrontation entre les deux schémas dimensionnels résultant de ces deux démarches. Cette confrontation est représentée dans la section suivante.

6. Confrontation

Après avoir conçu le schéma dimensionnel des besoins en suivant la démarche descendante et le schéma dimensionnel de l'entreprise en se basant sur l'entrepôt de données, nous procédons à l'intégration de ces deux schémas. Cette fusion est réalisée en confrontant les deux schémas afin d'enlever, d'ajouter ou de purifier quelques informations. Il faut noter que cette étape d'intégration nécessite l'intervention de toute l'équipe du décisionnel : informaticiens et décideurs. L'intervention des décideurs au niveau de l'intégration est primordiale afin de tester la complétude du schéma dimensionnel.

La confrontation est basée sur deux étapes :

- la première étape consiste à construire un schéma dimensionnel qui comporte les données communes : fait, dimension, hiérarchies, … ;
- la deuxième réalise l'analyse des deux schémas dimensionnels confrontés pour intégrer les informations jugées pertinentes par l'ensemble des informaticiens et des décideurs.

Ainsi, l'intégration des données des deux schémas permet :

- *la définition correcte de la granularité de l'analyse*. Le choix d'un niveau de détail est très important dans la conception d'une base dimensionnelle. Le choix d'une granularité trop fine risque d'augmenter la taille de la base dimensionnelle en dérivant à partir de l'entrepôt des données détaillées non pertinentes à l'analyse. Par contre, le choix d'une granularité moins fine ne permet pas d'analyser les données les plus détaillées. A ce niveau, le concepteur doit faire appel aux décideurs pour confronter leurs besoins exprimés au niveau de la démarche descendante aux granularités d'analyse détaillées obtenues lors de la démarche ascendante. Cette confrontation permet de découvrir des besoins non déclarés par les décideurs afin de déterminer une granularité adaptée aux besoins décisionnels ;
- *l'épuration des données*. En effet, les données extraites à partir des requêtes utilisateurs peuvent ne pas exister dans notre entrepôt de données. Dans ce cas, nous avons le choix entre enlever ces données ou bien les garder avec des valeurs vides en attendant d'avoir les informations nécessaires à leur instanciation.
- *ajout des données sources*. C'est le concepteur qui analyse les données de l'entrepôt et détecte les informations susceptibles d'intéresser le décideur. Souvent, ce dernier ne connaît pas le contenu de l'entrepôt ni les informations susceptibles d'être analysées. Nous pouvons, par exemple, ajouter des paramètres d'analyse en détectant de nouvelles propriétés dans les dimensions définies dans notre exemple de schéma dimensionnel ;
- *ajout des mesures calculées demandées par le décideur*. Ces mesures nécessitent l'utilisation d'une règle de gestion qui permet de calculer la mesure à partir d'un ensemble de données entrepôt ;
- *intégration de toutes les contraintes sémantiques*. Ces contraintes proviennent soit de la démarche descendante, soit de la démarche ascendante et détectées lors de l'analyse de l'entrepôt.

Exemple 17

Lors de la confrontation des deux schémas que nous avons obtenus à partir des démarches descendante et ascendante, nous avons été amenés à réaliser les transformations suivantes :

- ajout du fait *Performance* qui n'a pas été considéré par le concepteur lors de l'analyse de l'entrepôt ;
- l'enrichissement des dimensions *Agences*, *Clients* et *Véhicules* avec de nouveaux paramètres et attributs faibles ;
- la validation des contraintes entre la hiérarchie de la géographie française et celle de la géographie américaine. De même, pour les contraintes entre la hiérarchie de la classification française des véhicules et celle adoptée par les agences américaines.

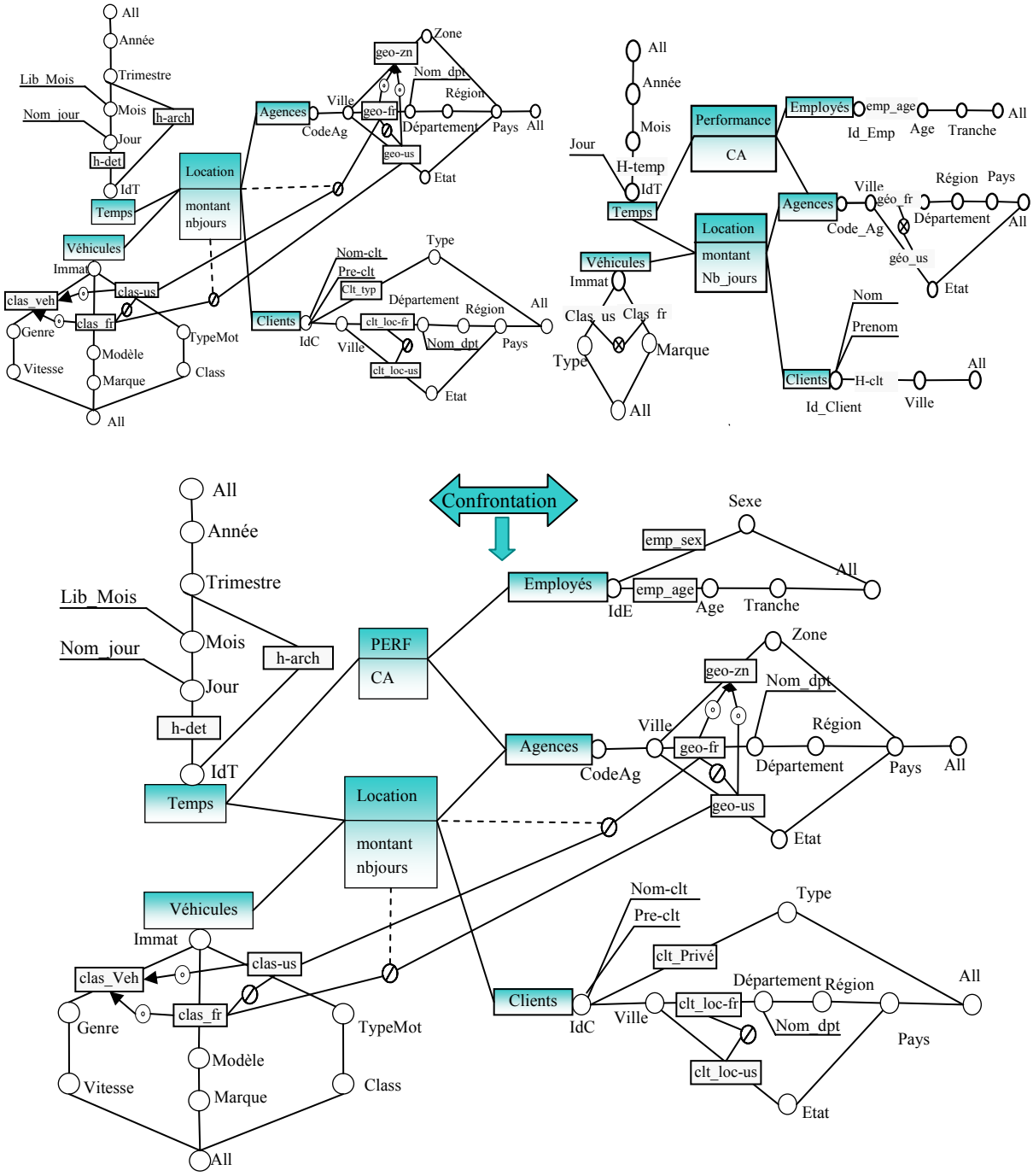


Figure IV.10 : *Exemple de confrontation des résultats des démarches ascendante et descendante*

La Figure IV.10 présente le résultat de cette confrontation qui est représenté par un schéma dimensionnel contraint.

Le concepteur doit familiariser les décideurs aux concepts du schéma dimensionnel et tester ensuite leur niveau de satisfaction en leur montrant les analyses décisionnelles possibles à travers ce schéma. Finalement, une étape de test auprès des décideurs est nécessaire afin de valider le schéma dimensionnel final.

7. Conclusion

L'objectif de ce chapitre est de proposer une méthode de conception d'une base de données dimensionnelles fiables et cohérentes intégrant toutes les données pertinentes à l'aide à la prise de décision. Pour répondre à cet objectif, nous proposons une méthode basée sur les niveaux d'abstraction : conceptuel, logique et physique. Ce chapitre est consacré à la présentation de la phase conceptuelle permettant de concevoir le schéma dimensionnel des données décisionnelles. Au niveau de cette phase, nous proposons une méthode mixte basée sur une démarche descendante et ascendante. Lors de la démarche descendante, nous proposons un ensemble d'étapes qui permettent de collecter les besoins des décideurs, de les spécifier et de les formaliser sous forme de schéma dimensionnel en constellation intégrant un ensemble de contraintes sémantiques. La démarche ascendante permet de collecter les données sources et de construire le schéma dimensionnel dédié à l'aide à la prise de décision. Une phase de confrontation est nécessaire afin de proposer un schéma dimensionnel intégrant les besoins des décideurs définis sans connaissance préalable des sources tout en tenant compte des données dérivées directement par le concepteur à partir de ces sources.

L'avantage de notre méthode est la combinaison des démarches ascendante et descendante permettant de tenir compte de toutes les informations pertinentes dans le processus décisionnel. Lors de la démarche descendante, nous portons un intérêt spécial à la phase de spécification des besoins des décideurs en faisant abstraction des sources de données. En outre, cette méthode (démarches descendante et ascendante) est basée sur un modèle dimensionnel intégrant un ensemble de contraintes favorisant la définition de bases de données dimensionnelles cohérentes et fiables.

En se basant sur les étapes de notre démarche ascendante, nous proposons d'automatiser la conception du schéma dimensionnel à l'aide de notre système d'aide à la conception de bases de données dimensionnelles. Cet outil est représenté dans le chapitre V.

CHAPITRE V : OUTIL D'AIDE A LA CONCEPTION DE MAGASIN DIMENSIONNEL CONTRAINT

PLAN DU CHAPITRE

Dans le cadre de la définition d'une méthode de conception de bases de données dimensionnelles complète, nous avons proposé :

- un modèle dimensionnel contraint et les formalismes graphiques associés (cf. Chapitre II),
- deux démarches de conception de schémas dimensionnels complémentaires : une démarche descendante basée sur les besoins des décideurs et une démarche ascendante partant du schéma de la source (cf. Chapitre IV § 4 et 5).

Pour compléter cette méthode et valider notre démarche ascendante (cf. Chapitre IV, section 4), nous proposons dans ce chapitre un outil d'aide à la conception. Cet outil permet une conception graphique et incrémentale de bases de données dimensionnelles.

Dans la première section, nous présentons les outils d'aide à la conception proposés dans la littérature et dans l'industrie. Dans la deuxième section, nous présentons l'architecture générale de l'outil et les notations graphiques adoptées pour la représentation de l'entrepôt et du magasin. La troisième section présente le référentiel de méta-données permettant de stocker les caractéristiques de notre modèle dimensionnel. La quatrième section présente notre langage graphique de conception de schéma dimensionnel basé sur une démarche ascendante incrémentale. La cinquième section est consacrée à la transformation logique de notre modèle dimensionnel selon un modèle ROLAP. Enfin, une dernière section présente nos modules de génération automatique du magasin dans un SGBD hôte comportant l'implantation du schéma, l'alimentation et le rafraîchissement des données.

1. Introduction

Notre outil, appelé GMAG (Générateur de MAGasins de données), est basé sur la dichotomie d'espaces de stockage, entrepôt et magasin de données, adoptée dans notre architecture de système décisionnel. Son rôle est d'assister le concepteur dans la définition du schéma dimensionnel du magasin à partir d'un entrepôt de données historisées. Le schéma dimensionnel obtenu est basé sur notre modèle en constellation intégrant l'expression de plusieurs faits analysés suivant des dimensions à multi hiérarchies. En outre, l'outil doit permettre au concepteur d'exprimer les contraintes sémantiques supportées par notre modèle dimensionnel.

Peu de travaux de recherche dans le domaine des systèmes décisionnels présentent un outil d'aide à la conception de schémas dimensionnels au niveau conceptuel. (Golfarelli et al, 2002) propose un outil de type CAISE[1], appelé WAND, basé sur un langage de définition de données graphiques opérant via l'intermédiaire d'éditeurs graphique de schémas dimensionnels. (Trujillo et al, 2002) propose une extension d'UML pour modéliser les bases dimensionnelles au travers d'un module d'extension de l'éditeur Rational Rose.

Ces outils ne prévoient pas une dichotomie entrepôt et magasins de données telle que nous la proposons dans cette thèse. Les solutions offertes dans ces travaux sont basées sur une source de données qui ne gère pas le temps ou bien qui ne conserve que l'historique détaillé des données. Or, dans notre architecture, l'entrepôt conserve l'historique des données sous forme détaillée et/ou résumée (archivée) (Ravat et al, 2000a). Notre modèle dimensionnel permet de restructurer ces données détaillées et archivées.

[1] *Computer Assisted Software Engineering, conception logicielle assistée par ordinateur.*

Dans le cadre industriel, le marché des applications décisionnelles rassemble plusieurs types d'outils tels que :

- les outils d'administration permettant le stockage et la gestion des données de l'entrepôt de données ;
- les outils de constitution (ETL) permettant d'extraire les données des bases de production, de les transformer et de les charger ;
- les outils de restitution rassemblant l'ensemble des outils utilisés pour l'analyse dimensionnelle des données (voir Annexe).

Ces logiciels bien que puissants et faciles d'utilisation, souffrent de l'absence de visualisation globale du schéma dimensionnel car ils se limitent à la présentation d'un unique schéma en étoile. Ils couvrent partiellement les étapes de conception d'une base dimensionnelle. Néanmoins, ces outils ne proposent pas, pour la plupart, une démarche claire et méthodologique pour la conception des magasins de données dimensionnelles.

2. L'outil GMAG

Nous proposons un outil de conception de schéma dimensionnel reposant sur des représentations graphiques de l'entrepôt et du magasin de données présentées dans les chapitres précédents. L'utilisation d'un langage graphique de conception offre une vision plus explicite de la réalité modélisée que celle offerte par les autres types de langages (textuels ou tabulaires) (Le Parc, 1997). En plus, ce langage permet de concevoir d'une manière incrémentale et progressive le schéma dimensionnel.

2.1. Architecture de GMAG

Dans cette section, nous présentons l'architecture générale de notre outil GMAG.

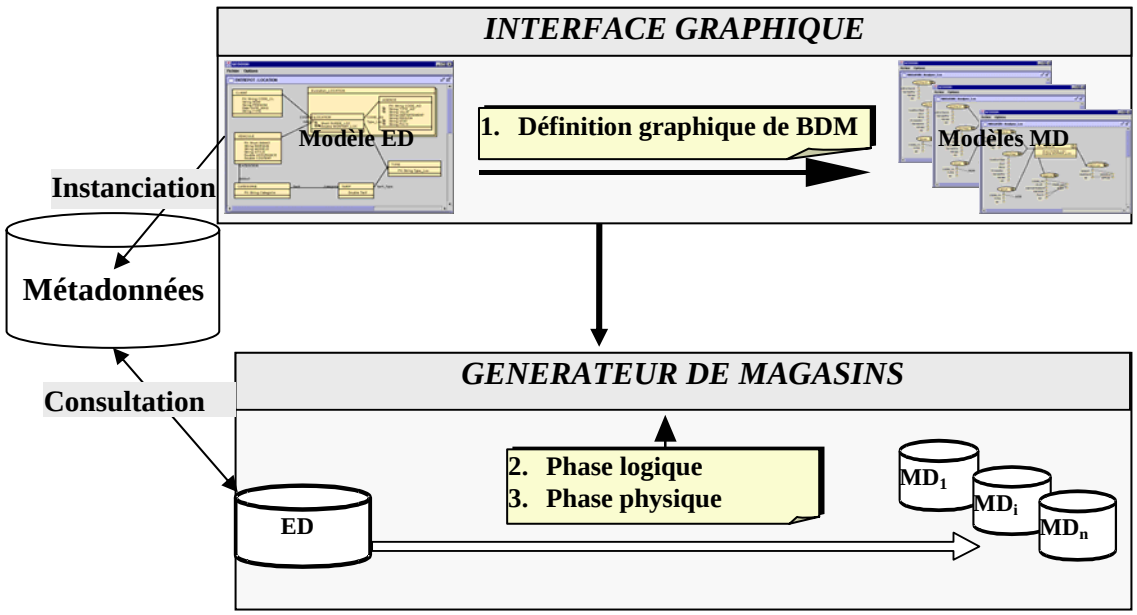


Figure V.1 : *Architecture du prototype GMAG*

Ce prototype comporte trois modules :

- ***le référentiel de méta-données*** décrit les structures de l'entrepôt et du magasin dimensionnel et la démarche de transformation entre ces deux espaces de stockage. Ce référentiel est instancié par l'interface graphique suite à la définition des différentes entités du schéma dimensionnel ;
- ***l'interface graphique*** permet au concepteur de définir graphiquement le schéma conceptuel dimensionnel à contraintes du magasin de données ;
- ***le module de génération*** permet d'implanter automatiquement les magasins de données dimensionnelles dans un SGBD relationnel.

2.2. Utilisation de GMAG

Au niveau de l'interface, le concepteur dispose d'une représentation graphique de l'entrepôt de données et d'une représentation graphique du magasin de données.

2.2.1. La fenêtre de l'entrepôt

L'entrepôt de données est représenté d'une manière graphique dans une fenêtre. Les notations utilisées reprennent celles du diagramme des classes UML étendu (Teste, 2000). L'extension des notations permet de représenter les concepts spécifiques à la modélisation de l'entrepôt tels que les environnements et les filtres des classes.

La fenêtre de l'entrepôt est composée de deux zones : une zone d'en-tête comportant le nom de l'entrepôt et le menu "***Fichier***" et une zone d'affichage du schéma de l'entrepôt (cf. Figure V.3). Le menu "Fichiers" permet d'ouvrir un entrepôt de données ou de sauvegarder sa représentation. Le schéma de l'entrepôt est constitué d'un ensemble de classes, d'environnements et de liens. Dans ce schéma, une notation spécifique est définie pour les environnements et les filtres :

- un environnement est représenté par un double rectangle contenant le nom de l'environnement et englobant les classes contenues dans l'environnement ;
- une propriété appartenant au filtre temporel d'une classe est précédée par un double rectangle blanc tandis qu'un attribut appartenant au filtre d'archives est précédé par un double rectangle noirci.

2.2.2. La fenêtre du magasin

Le magasin de données est lui aussi représenté de façon graphique dans une fenêtre. Les notations utilisées sont fidèles au formalisme graphique de notre modèle dimensionnel contraint présenté dans le deuxième chapitre (cf. chapitre II, section 2).

La fenêtre du magasin est composée de deux zones : une zone d'en-tête, comportant le nom du magasin et les menus des opérations, permettant au concepteur de construire le magasin, et une zone d'affichage du schéma du magasin (cf. Figure V.15).

Le menu "***Fichiers***" permet d'ouvrir un ancien magasin, de créer un nouveau ou de sauvegarder sa représentation. Le menu "***Magasin***" comporte l'ensemble des opérations permettant la définition du schéma dimensionnel à partir du schéma de l'entrepôt. Il comporte les sous menus suivants :

- **Dérivation** : permet de dériver un fait ou une dimension.
- **Historisation :** permet de définir les dimensions temporelles.
- **Configuration :** permet de relier les faits aux dimensions, de définir les hiérarchies et d'exprimer les contraintes.
- **Génération :** permet de générer le magasin de données ROLAP dans un SGBD relationnel**.**
- **Interrogation :** permet de visualiser les données des faits et des dimensions**.**

Dans le schéma du magasin, nous adoptons les notations suivantes :

- un ***fait*** est représenté par un rectangle divisé en deux parties : une en-tête comportant le nom du fait et un corps comportant les mesures du fait ;
- une ***dimension*** est représentée par un rectangle comportant le nom de la dimension.
- Un ***paramètre*** est représenté par un cercle étiqueté par le nom du paramètre.
- Un ***attribut faible*** est représenté par son nom souligné et relié au paramètre qu'il décrit.

La description des structures et des contraintes du schéma dimensionnel est stockée dans un référentiel de méta-données afin de pouvoir les utiliser par la suite lors de la phase physique. Une description détaillée de ce référentiel fait l'objet de notre prochaine section.

3. Le référentiel des méta-données

Pour décrire les structures de notre base de données dimensionnelles, nous avons conçu un méta-modèle UML (cf. Figure V.2) qui permet de décrire les caractéristiques du schéma dimensionnel du magasin défini par le concepteur.

Ce diagramme de classes UML est utilisé par notre interface graphique afin de conserver l'historique des opérations réalisées par le concepteur pour créer son schéma dimensionnel. Les objets des classes de notre référentiel sont, donc, les faits, les dimensions et les hiérarchies d'un schéma dimensionnel définis par le concepteur. Le choix du modèle orienté objet pour la représentation du niveau méta est justifié par la riche sémantique de ce modèle et le support de puissants concepts tels que l'héritage et la composition utilisés souvent au niveau méta (Muller, 2000).

Dans cette section, nous présentons les classes principales qui composent notre méta-modèle de base de données dimensionnelles.

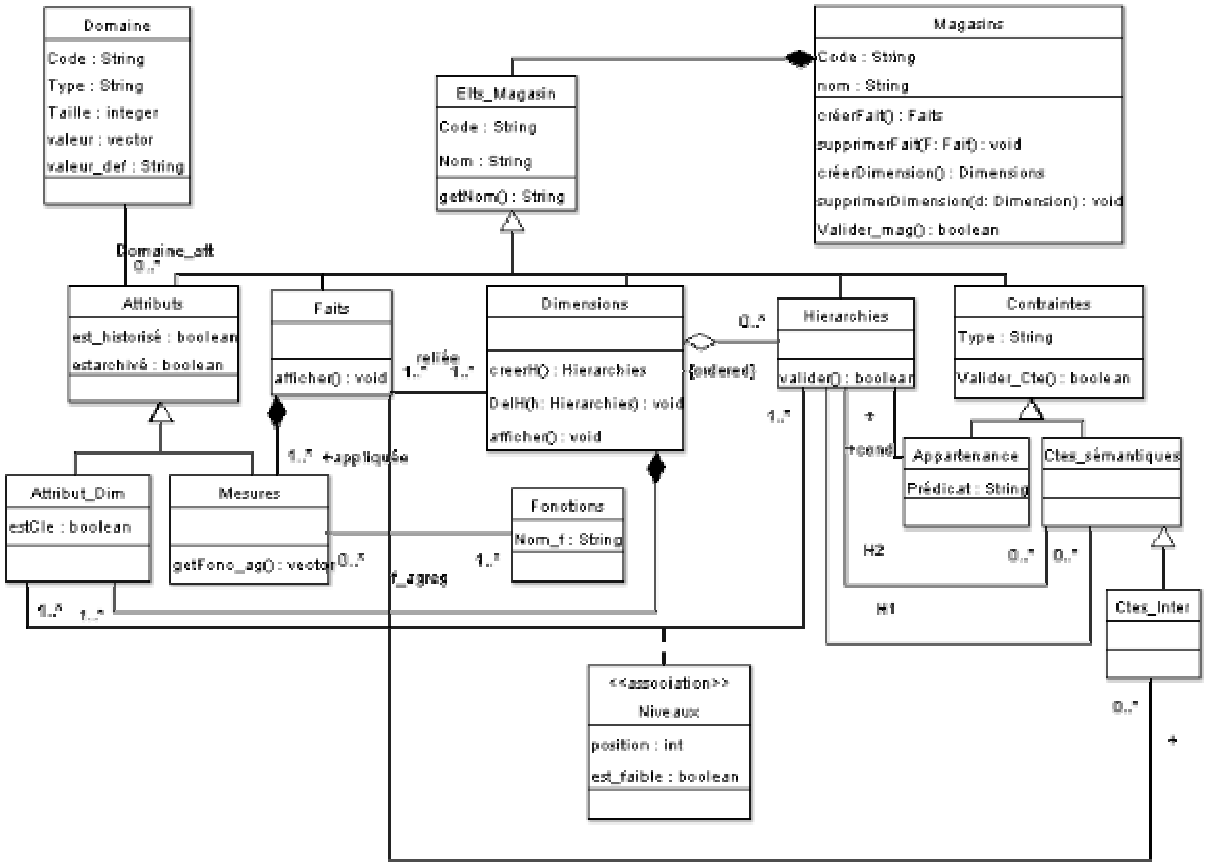


Figure V.2 : *Diagramme de classes UML simplifié du référentiel de méta-données*

A partir de l'entrepôt, plusieurs magasins dimensionnels sont concevables selon les besoins des utilisateurs. Ce concept est représenté dans le méta-modèle par la classe ***Magasin.*** Un objet de cette classe représente un magasin. Un magasin est caractérisé par un code et un nom. La classe ***Magasin*** est reliée par des liens de composition à la classe ***Elt_Magasin*** comportant les faits, les dimensions, les attributs, les hiérarchies et les contraintes.

La méta-classe ***Elt_Magasin*** représente le concept abstrait d'un élément du magasin qui rassemble les concepts de faits, de dimensions, de hiérarchies, d'attributs et de contraintes.

La méta-classe ***Faits*** représente le concept de fait. Elle décrit les mesures d'activité de la base de données dimensionnelles. Chaque mesure d'activité est une propriété du fait. Un fait possède un ensemble de dimensions qui le paramètrent.

La méta-classe ***Dimensions*** représente le concept de dimension. Elle décrit les paramètres de l'analyse. Chaque paramètre est un attribut de la dimension. Cette classe est liée à l'ensemble des faits dimensionnés ; dans notre modèle en constellation, une dimension peut être partagée entre plusieurs faits.

La méta-classe ***Attributs*** comporte les attributs des faits et des dimensions. Elle hérite de la méta-classe ***Elt_magasin***. Les attributs peuvent représenter les mesures, les paramètres de l'analyse ou les attributs faibles associés aux paramètres. Chaque attribut possède un domaine. Dans notre modèle dimensionnel, nous proposons de conserver l'historique des données au niveau détaillé et/ou archivé. Les deux propriétés booléennes *est_historisé* et *est_Archivé* de cette classe permettent de caractériser les attributs dont les données seront historisées. Cette information est nécessaire lors de la transformation du schéma conceptuel

dimensionnel en schéma logique, afin d'ajouter les attributs temporels 'début' et 'fin' de transaction.

La méta-classe ***Mesures*** rassemble les mesures d'activités contenues dans les faits. Elle hérite de la classe ***Attribut***. Chaque mesure possède un ensemble de fonctions d'agrégation qui permettent de résumer sa valeur lors du passage d'un niveau de détail à un autre au niveau d'une dimension.

La méta-classe ***Attributs_Dim*** rassemble les paramètres d'activités définies dans les dimensions et les descripteurs de ces paramètres appelés *attributs faibles*. Chaque dimension possède un paramètre clé. Ces paramètres sont organisés en niveaux hiérarchiques.

La méta-classe ***Hiérarchies*** représente le concept de hiérarchie. Elle permet d'organiser les différents attributs des dimensions en structure hiérarchique à l'aide de la méta-classe d'association ***Niveaux***. En effet, une hiérarchie est composée d'un ensemble de niveaux correspondant aux attributs dimensionnels. Un niveau, caractérisé dans la hiérarchie par sa position, comporte un attribut de base appelé paramètre et un ensemble d'attributs faibles.

La méta-classe ***Contraintes*** représente les contraintes appliquées sur les données dimensionnelles du magasin. Ces contraintes permettent de conserver l'intégrité et la cohérence de la base dimensionnelle lors de sa construction et de mieux visualiser ses données pendant l'étape d'interrogation. Nous distinguons, à ce niveau, les contraintes sémantiques intra et inter dimensions de la contrainte d'appartenance définie sur les hiérarchies pour définir les instances qui lui appartiennent.

Ce méta-modèle stocke les caractéristiques du magasin de données dimensionnel contraint définies par le concepteur. Ce dernier définit graphiquement les schémas conceptuels au travers d'interfaces (cf. Figure V.1). Cette définition graphique est étudiée dans la section suivante.

4. Définition graphique d'un magasin de données dimensionnel contraint

Nous proposons une démarche graphique et incrémentale de conception des magasins de données. En effet, le module interface de notre prototype permet à l'administrateur chargé de la construction d'un magasin de visualiser le schéma de l'entrepôt (Figure V.3) et de réaliser progressivement les opérations de modélisation et de configuration du magasin. Le magasin résultant de cette construction est basé sur un modèle dimensionnel contraint.

Nous avons proposé dans le chapitre IV une démarche de conception ascendante de magasin de données comportant six étapes :

1) Détermination des faits représentant les sujets analysés.
2) Détermination des dimensions représentant les perspectives de l'analyse.
3) Définition de la granularité des données de l'analyse.
4) Organisation des paramètres des dimensions selon des dépendances hiérarchiques pour supporter les analyses à différents niveaux de détail.
5) Définition de la dimension temporelle et détermination des faits qui seront reliés aux hiérarchies temporelles détaillées et/ou d'archives.
6) Expression des contraintes sémantiques inter et intra-dimension.

Nous présentons dans cette section l'implantation de cette démarche à l'aide de notre outil graphique. Les différentes étapes sont illustrées par des exemples basés sur l'entrepôt de données présenté dans le paragraphe suivant.

4.1. Exemple d'un entrepôt historisé

Notre démarche de conception de magasin se base sur un entrepôt de données historisées (Ravat et al, 2000b). Nous rappelons dans cette section notre exemple d'entrepôt de données historisées présenté dans le chapitre IV. Dans notre interface, une fenêtre spécifique représente le schéma de l'entrepôt de données selon le diagramme de classes UML étendu.

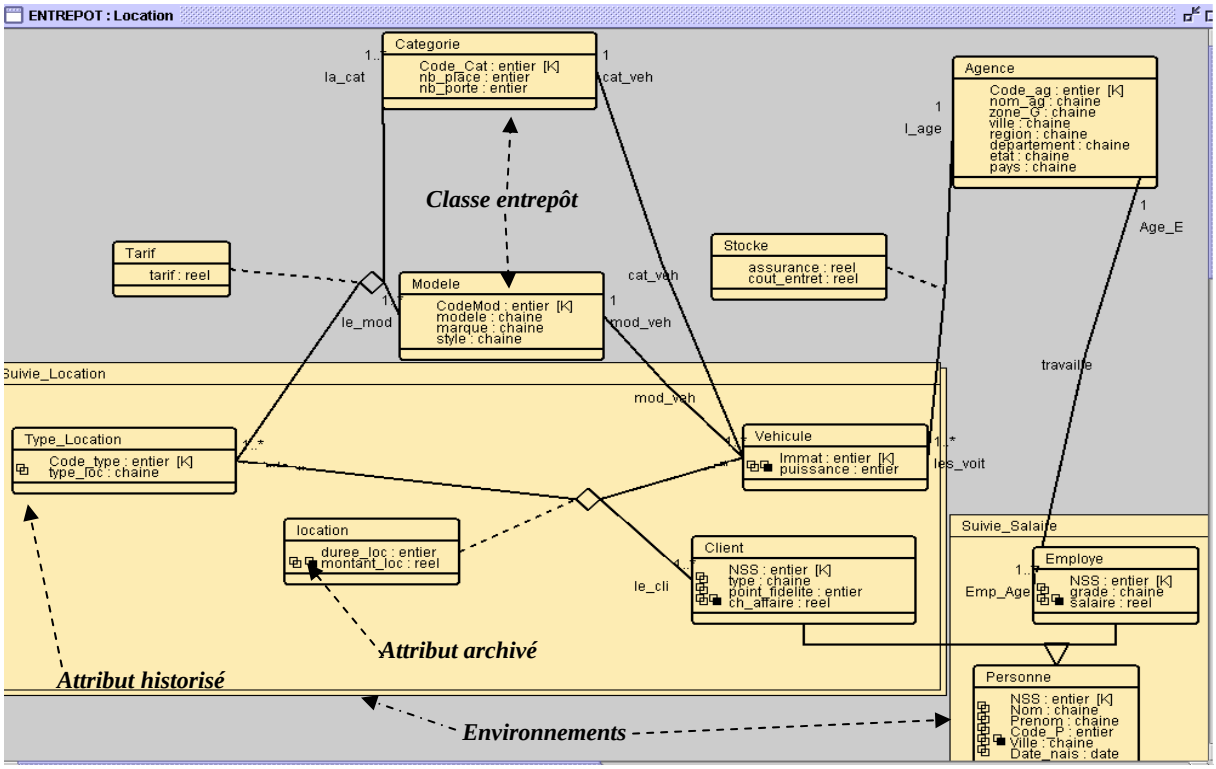


***Figure V.3**: Schéma de l'entrepôt de données selon le diagramme de classes UML étendu.*

Notre exemple présente une application de gestion de location de véhicules d'une société qui possède plusieurs agences en France et aux Etats-Unis.

Afin de conserver l'historique des locations journalières et de réaliser le suivi mensuel des salaires des employés, nous avons défini les environnements 'Suivi_Location' et 'Suivi_Salaire'. L'environnement 'Suivi_Location' comporte les classes *Client*, *Type_Location*, *Véhicule* et *Location*. L'environnement 'Suivi_Salaire' comporte les classes *Personne* et *Employé*.

A partir de ce schéma d'entrepôt, le concepteur définit les différentes composantes de son schéma dimensionnel en se basant sur la démarche ascendante définie dans le chapitre précédent.

4.2. Détermination des faits

Dans un premier temps, le concepteur détermine les classes de l'entrepôt représentatives des sujets d'analyse traités dans le magasin. Selon notre démarche, une classe de fait est

projetée à partir d'une classe représentative choisie par le concepteur. Afin de créer un fait, le concepteur sélectionne la classe représentative dans la fenêtre entrepôt et choisit l'opération "*Dériver fait …"* du sous menu "*Dérivation*" du menu "*Magasin*". Cette opération permet d'afficher une fenêtre de dialogue qui comporte le nom du fait et l'ensemble des attributs de la classe représentative sous forme de cases à cocher. À la fin de cette opération, l'outil demande au concepteur l'ensemble des fonctions d'agrégation compatibles avec chaque mesure du fait. La validation de l'opération crée un nouveau fait et l'affiche dans la fenêtre du magasin. La création du fait est accompagnée par une opération de mise à jour du référentiel de méta-données.

Exemple 1

La définition du fait Loc_Vehicule, présenté dans la Figure V.4, est réalisée à l'aide de l'interface de la manière suivante : le concepteur sélectionne la classe entrepôt *Location* représentative du fait et choisit dans le menu l'opération de dérivation (Etape (1)). L'interface lui présente automatiquement les différents attributs de la classe *Location* sous forme de cases à cocher afin qu'il projette les indicateurs d'analyse adéquats (Etape (2)). Au niveau de l'étape (3), le concepteur choisit les fonctions d'agrégation compatibles avec chaque mesure. Enfin, la validation de ces opérations permet de créer le fait *Location* représenté dans l'étape (4)

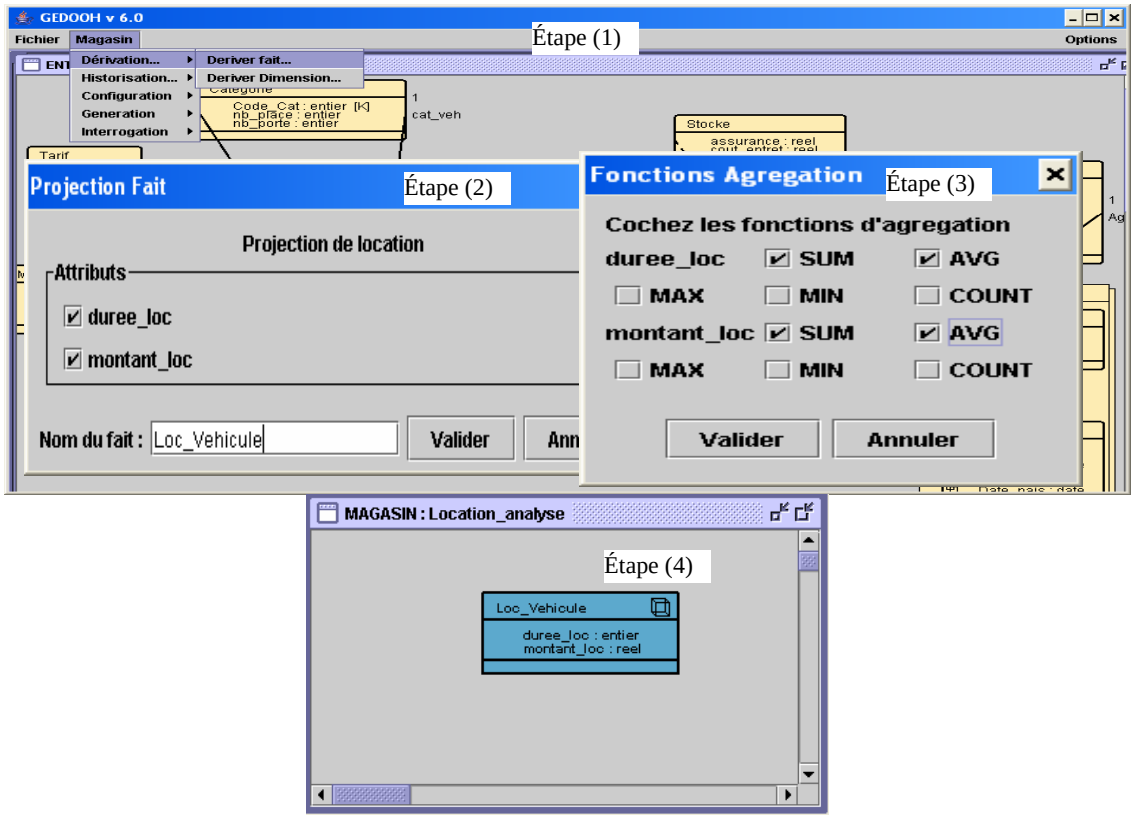


Figure V.4 : *Dérivation du fait « Loc_vehicule »*

4.3. Détermination des dimensions

Les dimensions sont élaborées à partir des classes entrepôts déterminantes de la classe représentative (autres que les dimensions temporelles). Une classe non déterminante ne peut pas dimensionner la classe représentative car elle ne détermine pas ses objets d'une manière

unique ce qui ne permet pas, par la suite, de paramétrer les mesures du fait issues de cette classe représentative. La liste des classes déterminantes d'une classe représentative est notée ***Determin (CR).***

La détermination de ces classes est réalisée automatiquement par notre outil en suivant le principe de dépendance fonctionnelle entre classes. Afin d'aider le concepteur lors de la sélection des différentes classes déterminantes, nous proposons un algorithme qui calcule automatiquement la liste de ces classes à partir d'une classe représentative donnée. L'algorithme que nous définissons dans la Figure V.5 présente les étapes de création d'une dimension et inclut la définition de la liste des classes déterminantes.

Algorithme Création Dimension
Entrées : `Classe représentative : CR`
Sorties : `Dimension créee.`
Debut
- *Déterminer la liste des classes entrepôt déterminantes de la classe représentative du fait CR* ***: Détermin (CR)*** *(voir algorithme suivant :* ***Algorithme recherche dépendance)****.*
- *Choisir les attributs A^{CD} de chaque classe déterminante à transformer en paramètres.*
- *Créer la classe de dimension comprenant les attributs A^{CD} dans le magasin.*
- *Affecter la classe de dimension à sa classe de fait.*

Fin

Algorithme recherche_dépendance
Entrées : `Classe représentative CR`
`Liste des classes traitées Cl_traitées.`
Sorties : `Liste des classes déterminantes de CR` **`Détermin(CR).`**
Debut

```
1.   Détermin (CR) = CR ;
// recherche dans la liste des super classes de CR
2.   Pour (i=0 ; i< taille(Super^CR) ; i++) Faire
3.      Sc ← Super^CR(i);
4.      Si (! Cl_traitées.contient(sc))  //la classe sc n'a pas été traitée  Alors
5.         Détermin (CR) ← recherche_dependance (sc, Détermin (CR)) ;
6.      Fin Si
7.   FinPour
// recherche dans les classes reliées par une relation de type 0..1.
8.   Pour (j=0 ; j< taille(Relations^CR); j++) Faire
9.      r ← Relations^CR( j) ;
10.     Si (Cardinalité(r)= «0..1» ) Alors
11.        Si ( ! Cl_traitées.contient(ClasseInverse(r)) ) Alors
12.           Détermin(CR)←recherche_dependance(ClasseInverse(r)), Détermin(CR));
13.        FinSi
14.     FinSi
15.  FinPour
16.  Retourner(Détermin(CR)).
```

FinRechercheDépendance

Super^{CR} *renvoie la liste des supers classes de la classe CR*
Relation^{CR} *renvoie la liste des relations de la classe CR*
Cardinalité(r) *renvoie la cardinalité de la relation r.*
ClasseInverse(r) *permet de récupérer la classe liée à la classe CR par la relation r.*
taille(liste) *permet de renvoyer la taille d'une liste de données.*

Figure V.5 *: Algorithme de définition d'une dimension et des classes déterminantes*

Exemple 2

L'application du principe de dépendance sur l'exemple précédent nous amène à déterminer l'ensemble des classes déterminantes de la classe représentative Location.

Détermin (*Location*) = {*Location, Véhicule, Agence, Client, Catégorie, Modèle, Personne, Type_Location*}.

Notre prototype permet au concepteur de visualiser automatiquement les classes déterminantes d'une classe représentative. Au niveau de l'interface, l'ensemble de ces classes change de couleur (grisées) pour permette au concepteur de choisir une classe à partir de laquelle il définira une dimension du fait (Figure V.6).

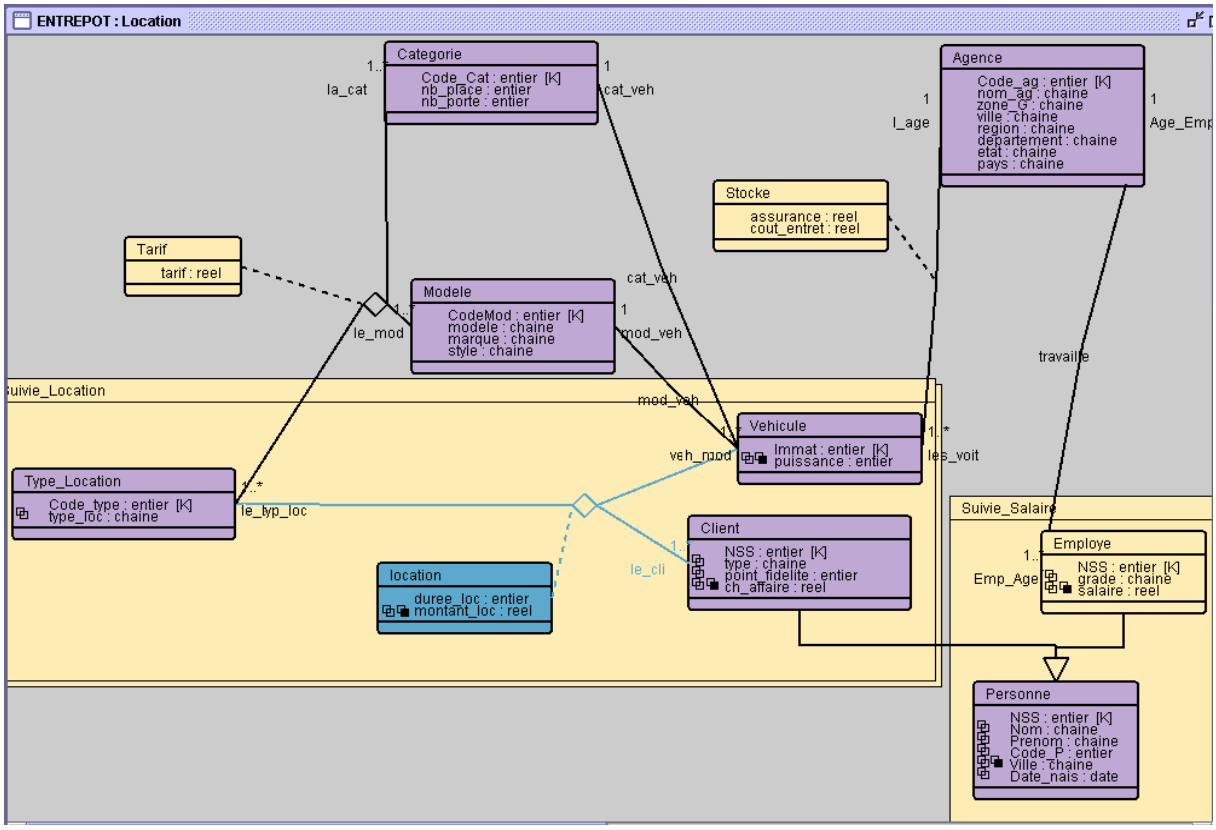


Figure V.6 : *Application du principe de dépendance*

A partir des classes déterminantes, les dimensions du fait *Location_Véh* peuvent être définies.

En se basant sur le résultat de notre méthode de conception présentée dans le chapitre IV, le concepteur choisit de définir les dimensions *Agence*, *Véhicule* et *Client* pour le fait *Loc_véhicule*.

La Figure V.7 présente l'exemple de définition graphique de la dimension *Agence*. Le concepteur sélectionne une classe parmi les classes déterminantes de la classe représentative du fait. Ensuite, il choisit dans le menu "*Magasin"* de la fenêtre du magasin, l'opération "*Dériver dimension"* (Etape (1)). Cette opération permet d'afficher une fenêtre de dialogue comportant les attributs de la classe déterminante sous forme de cases à cocher. Le concepteur choisit les attributs qu'il souhaite dériver dans sa nouvelle dimension, définit le nom de cette dernière et valide l'opération (Etape (2)).. La validation

génère automatiquement dans la fenêtre du magasin le schéma graphique de la dimension *Agence* (Etape (3)). Les attributs de cette dimension sont tous attachés à la clé de celle-ci vu que les hiérarchies ne sont pas encore définies.

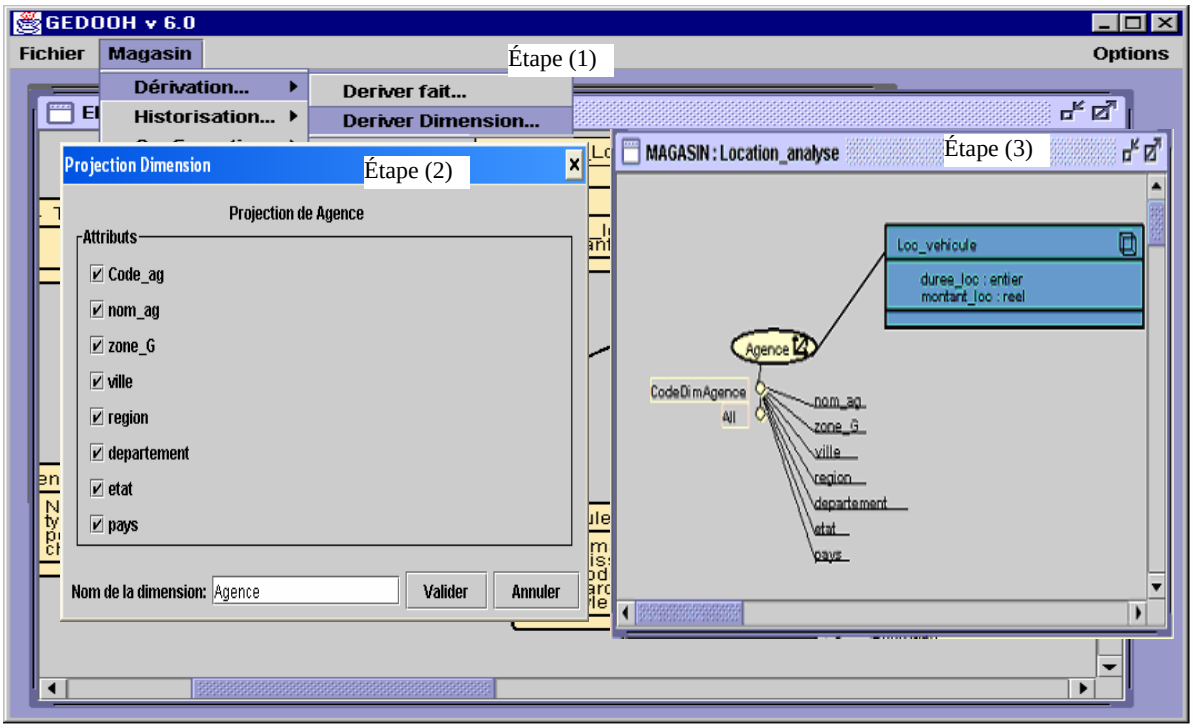


Figure V.7 : *Dérivation de la dimension Agence*

4.4. Hiérarchisation des dimensions

Les paramètres des dimensions définies ci-dessus respectent une structure hiérarchique qui permet d'indiquer le niveau de détail de l'analyse. En effet, les mesures d'activité sont observées selon plusieurs perspectives et à différents niveaux de granularité. La définition d'une hiérarchie est réalisée par la détection des dépendances hiérarchiques entre les paramètres d'une dimension. Cette détection peut être réalisée automatiquement en analysant les valeurs des paramètres au niveau de l'entrepôt. Ainsi, nous proposons de définir pour chaque paramètre p_i la liste des paramètres qui le suit dans la hiérarchie, ***param(p_i)***.

Algorithme DépendanceHiérarchique
Entrées : *ListeP : liste des paramètres de la dimension.*
R : Relation comportant les instances des paramètres de la liste analysée.
Sorties : *ListePara est une matrice dont chaque ligne i comporte les paramètres qui le suivent dans les hiérarchies.*
Debut
1. ***Etape1 :*** *Détection des dépendances fonctionnelles entre les paramètres*
2. *Pour tout couple de paramètres (p_i, p_j) de ListeP Faire*
3. *param=vrai ;*
4. *Pour chaque instance Ipi $\in$ **$\Pi_{pi}(R)$** Faire*
5. *Si **Cardinalité ($\delta_{pi= Ipi}$ ($\Pi_{pi, pj}(R)$))** <> 1 Alors*
6. *param= faux **;***
7. *FinSi*
8. *FinPour*
9. *Ajouter p_j à ListePara(i)= ;*
10. *FinPour*
11. ***Etape 2 :*** *simplification des dépendances : suppression des redondances*
12. *Pour chaque ligne i de ListePara Faire*
13. *Pour chaque couple (p_n, p_m) de ListePara(i) Faire*
14. *Si $p_m \in Param(p_n)$ Alors*
15. *supprimer p_m de ListePara(i) ;*
16. *FinSi*
17. *FinPour*
18. *FinPour*
19. *Retourner (ListePara)*

FinDépendanceHiérarchique

$\Pi_{pi}(R)$: *projette les valeurs du paramètre p_i dans la relation R*
$\delta_{pi= Ipi}(R)$: *sélectionne les instances de R qui vérifie la contrainte $p_i = I_{pi.}$*

Figure V.8 *: Algorithme de définition des dépendances hiérarchiques.*

L'algorithme de la Figure V.8 permet de construire une matrice, ***ListePara***, dont chaque ligne i comporte les paramètres ***param(p_i)*** succédant p_i dans la structure hiérarchique de la dimension. Nous présentons dans la figure suivante un exemple d'exécution de cet algorithme.

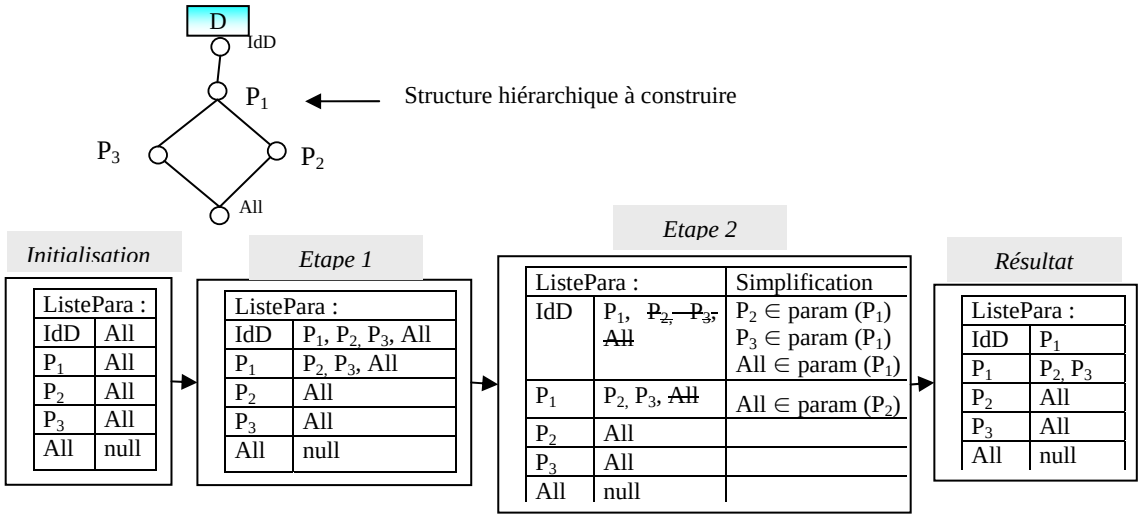


Figure V.9 *: Exemple d'exécution de l'algorithme DépendanceHiérarchique*

Afin de définir les hiérarchies multiples, nous proposons au concepteur une fenêtre spécifique à la création des hiérarchies qui s'affiche suite à la sélection d'une dimension dans la fenêtre du *magasin* et du choix de l'opération '*Construire hiérarchies…*' dans le sous-menu '*Configuration'* du menu '*Magasin'* (Figure V.10). Cette fenêtre comporte l'ensemble des attributs de la dimension que le concepteur peut réorganiser en listes correspondant aux différentes hiérarchies à définir. Pour chaque liste créée, le concepteur affecte un nom de hiérarchie. L'outil propose au concepteur les listes de paramètres correspondantes aux hiérarchies obtenues à partir de l'algorithme ***DépendanceHiérarchique*** et c'est le concepteur qui choisit de créer ou non ces hiérarchies. Nous notons que l'algorithme de définition des hiérarchies est basé sur les valeurs des paramètres qui peuvent être erronées (faute de frappe, valeur vide). Le résultat de cet algorithme est proposé au concepteur pour l'aider à définir les hiérarchies, mais c'est à lui de décider de la structure finale de la dimension.

Exemple 3

L'application de l'algorithme de détection des hiérarchies sur l'ensemble des paramètres de la dimension *Agence* (Etape (1)), nous a permis de définir l'ensemble des hiérarchies suivant :

- *geo_fr* = ('géo. française', Code_Ag, Ville, Département, Région, Pays, All),
- *geo_us* = ('géo. américaine', Code_Ag, Ville, Etat, Pays, All),
- *geo_zn* = ('zone agence', Code_AG, zone, All).

Suite à la validation du nom de chaque hiérarchie (Etape (2)), la fenêtre graphique du magasin est mise à jour automatiquement (Etape (3)).

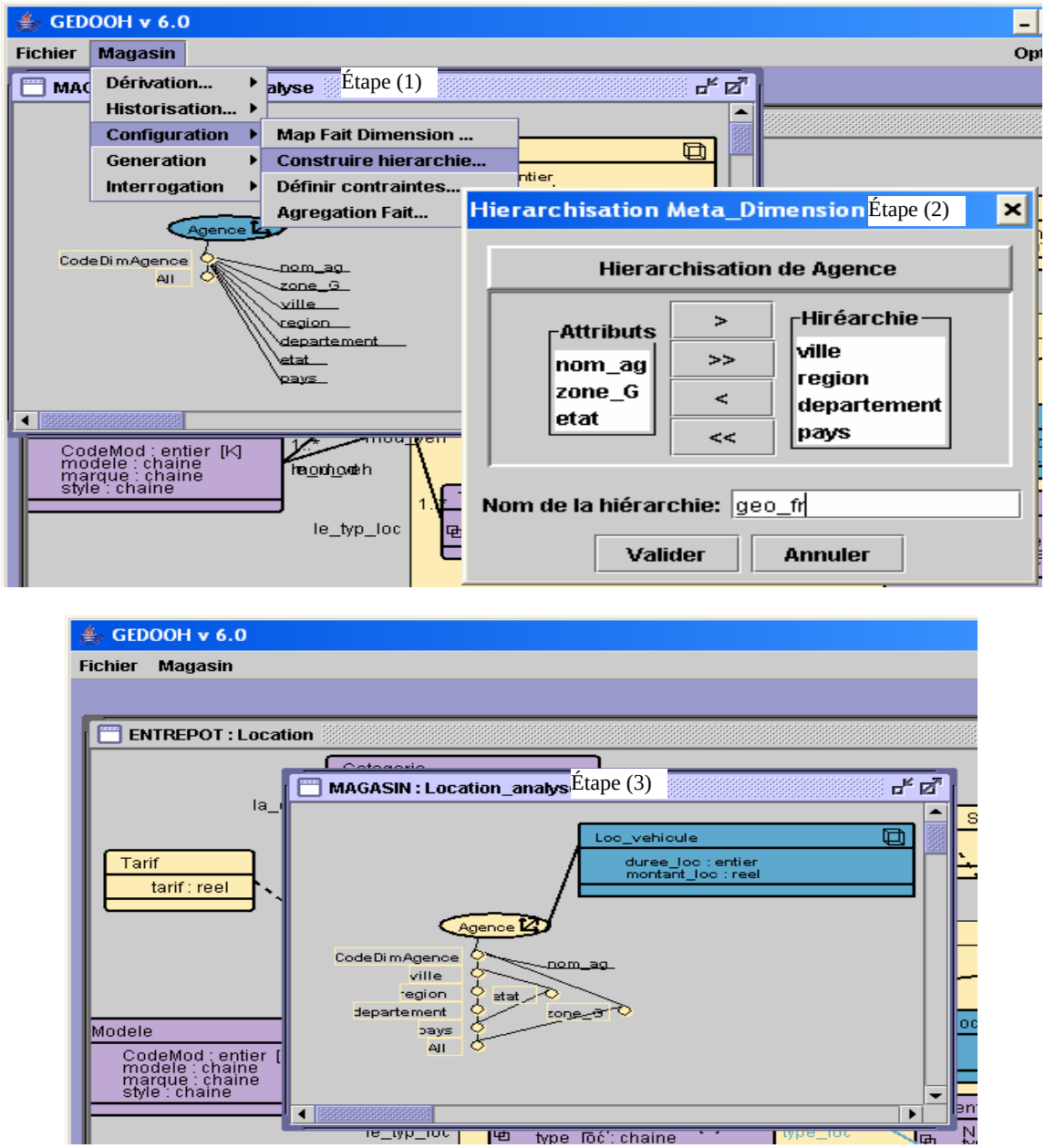


Figure V.10 : *Construction des hiérarchies de la dimension Agence*

4.5. Définition de la dimension temporelle

La dimension temporelle comporte deux hiérarchies permettant de décrire les données détaillées et les données agrégées telles qu'elles sont organisées dans l'entrepôt. La définition de cette dimension dépend de la configuration des environnements de l'entrepôt de données, des filtres temporels et des filtres d'archives de la classe représentative et des classes déterminantes. Afin d'aider le concepteur du magasin à définir ces hiérarchies, nous avons intégré dans l'outil un module d'analyse des classes représentatives et déterminantes. Ce module permet de détecter le niveau de granularité temporel défini au niveau de l'entrepôt. Ce niveau représente le niveau le plus détaillé de la hiérarchie temporelle détaillée. Le module permet d'autoriser la création d'une hiérarchie d'archives si les données du fait sont archivées au niveau de l'entrepôt.

Au niveau de notre interface, nous visualisons la dimension *Temps* d'une manière particulière. En effet, les deux hiérarchies appelées '*DimTDet*' et '*DimTArch*' sont affichées séparément. Le fait est connecté à ces deux hiérarchies par un arc commun qui se dissocie en deux branches reliées à chacune d'elles (cf. Figure V.15).

L'exemple suivant illustre cette étape de définition de la dimension temporelle.

Exemple 4

Dans l'exemple de notre magasin '*Location_analyse*', le concepteur souhaite analyser les données des locations à deux niveaux, détaillé et agrégé.

Pour réaliser cette analyse, il sélectionne le fait *Location_Veh* et choisit l'opération '*Détaillée*' dans le sous-menu '*Historisation*' du menu '*Magasin*' (Etape (1)). La fenêtre correspondante à l'historisation détaillée est automatiquement affichée avec le paramètre *jour* comme granularité la plus détaillée (Figure V.11 Etape (2)). Les autres attributs temporels sont affichés sous forme de cases à cocher. Le concepteur choisit les différents paramètres temporels nécessaires à l'analyse en se basant sur l'analyse des besoins réalisée au niveau de notre méthode de conception.

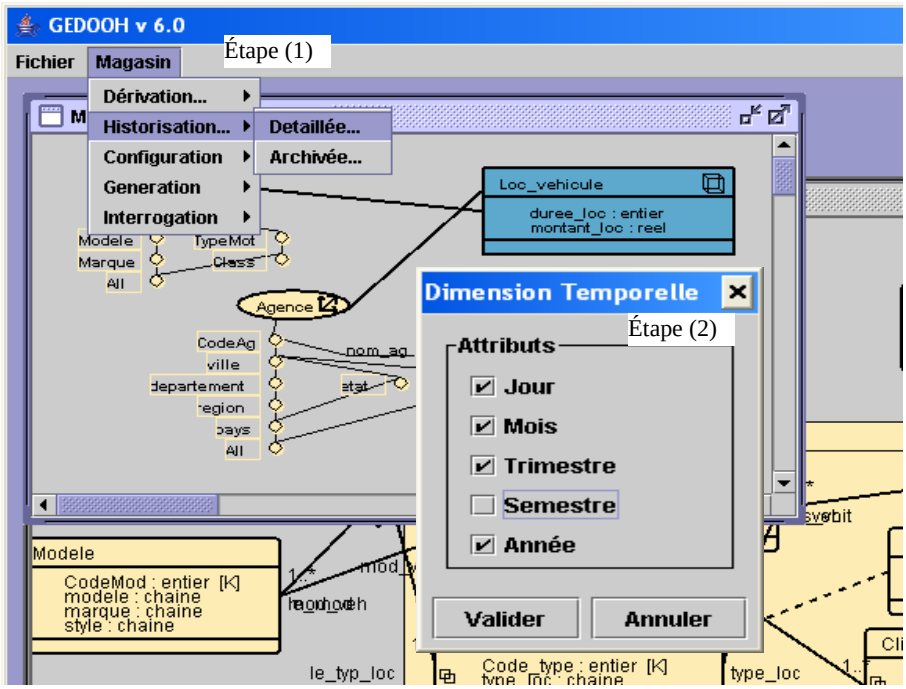


Figure V.11 : *Création de la hiérarchie temporelle détaillée*

L'analyse des données historisées correspondantes aux montant et durée de locations est réalisée en fonction des paramètres de la hiérarchie d'archives. Pour créer cette hiérarchie, le concepteur sélectionne l'option '*Archivée*' du sous-menu '*Historisation*' du menu '*Magasin*' (Figure V.12 Etape (1)). Les données des locations qui sont archivées tout les deux mois durant la période antérieure à l'année 1995 sont alors analysées en fonction de cette hiérarchie. La granularité la plus fine de cette hiérarchie est égale à deux mois. Elle est définie automatiquement par le système. Le concepteur choisit de prendre la granularité *Semestre* pour l'analyse de ces données (Etape (2)).

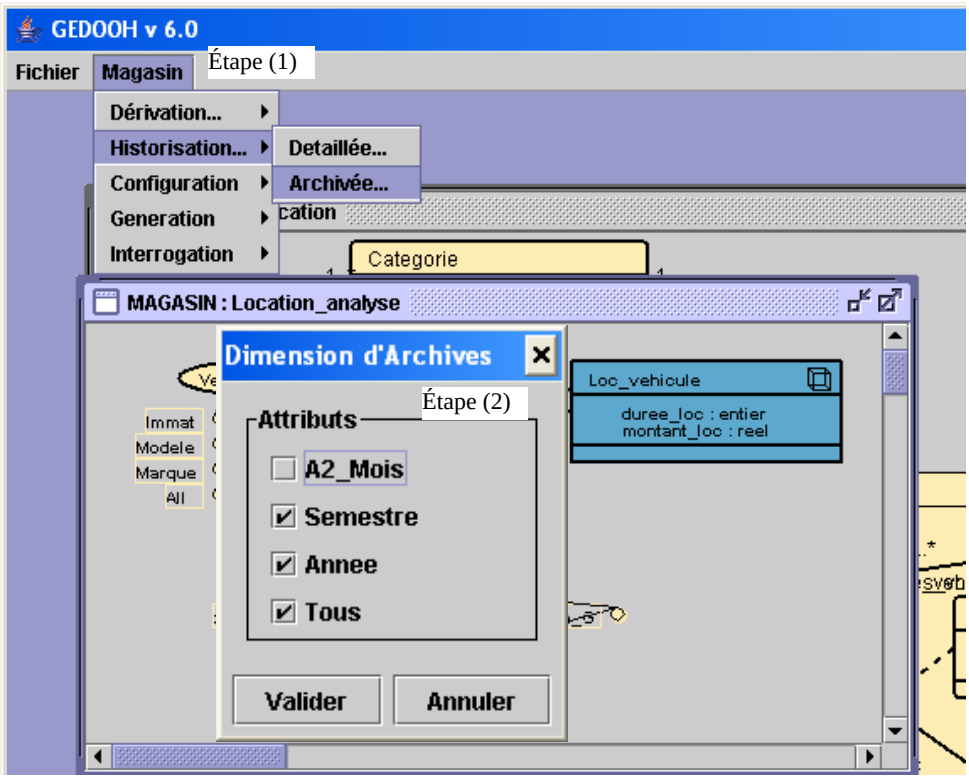


***Figure V.12** : Création de la hiérarchie temporelle archivée*

4.6. Expression des contraintes

Nous avons intégré dans notre modèle dimensionnel des contraintes sémantiques qui permettent de résoudre les conflits entre les instances des dimensions et de mieux organiser ces instances dans les différentes hiérarchies. Nous définissons des contraintes à deux niveaux :

– les contraintes intra-dimensions sont des contraintes qui concernent une seule dimension, autrement dit, il s'agit de contraintes entre les hiérarchies d'une même dimension ;

– les contraintes inter-dimensions sont des contraintes portant sur des hiérarchies de dimensions différentes.

4.6.1. Contraintes intra-dimensions

Nous rappelons dans ce qui suit les différentes contraintes qui peuvent êtres définies au niveau d'une dimension : *exclusion, inclusion, partition, simultanéité* et *totalité*.

Lors de la démarche ascendante, le concepteur définit les contraintes sémantiques appliquées sur les hiérarchies de chaque dimension en se basant sur l'analyse des domaines des paramètres d'analyse (cf. Chapitre IV § 5.5). Ces contraintes sont intégrées dans le schéma dimensionnel à l'aide de notre interface graphique (voir Figure V.13). Le concepteur sélectionne la dimension sur laquelle il va appliquer des contraintes. Ensuite, il choisit l'option '*Définir contraintes…*' dans le sous-menu '*Configuration*' du menu '*Magasin*'. Le choix de cette option lui affiche une fenêtre qui comporte deux listes de boutons radio correspondantes aux différentes hiérarchies de la dimension. Pour chaque hiérarchie sélectionnée l'outil affiche le graphe afin de rappeler sa structure au concepteur. Au milieu de ces deux listes, nous proposons une troisième liste de choix qui comporte les types de

contraintes que nous définissons dans notre modèle dimensionnel. La définition d'une contrainte entre deux hiérarchies déclenche une opération de mise à jour du référentiel des méta-données pour stocker cette nouvelle contrainte.

Exemple 5

Dans l'exemple de la dimension *Agence*, nous ne pouvons pas définir les paramètres *Département* et *Région* de la hiérarchie '*geo_fr'* pour les agences situées dans les différents états (organisées suivant la hiérarchie '*geo-us'*).

Par contre, nous remarquons que les agences situées dans les différents états (suivant la hiérarchie '*geo_us'*) possèdent des types et peuvent être visualisées selon la hiérarchie H_{type}. De même pour les agences décrites par la hiérarchie '*geo_fr'*.

Les hiérarchies '*geo_fr'* et '*geo_us'* de la dimension *Agence* contiennent la totalité des agences de la dimension. Ceci implique que chaque agence est décrite dans l'une ou l'autre des hiérarchies.

Ces contraintes sont définies à l'aide de l'interface graphique. Le concepteur sélectionne la dimension *Agence* comportant les hiérarchies concernées par la contrainte (cf. Figure V.13, Etape (1)). Dans la fenêtre de création des hiérarchies, nous visualisons les trois hiérarchies de la dimension *Agence* sous forme de bouton radio à droite et à gauche de la fenêtre (cf. Figure V.13, Etape (2)). Dans notre exemple, le décideur a sélectionné les hiérarchies '*geo_fr'* dans la première liste des hiérarchies et '*geo_us'* dans la deuxième liste des hiérarchies. Il a choisi, ensuite, la contrainte de totalité dans la liste des contraintes placée entre les deux hiérarchies. La validation de l'opération permet d'ajouter la contrainte de totalité à la liste des contraintes déjà définie, affichée en bas de la fenêtre.

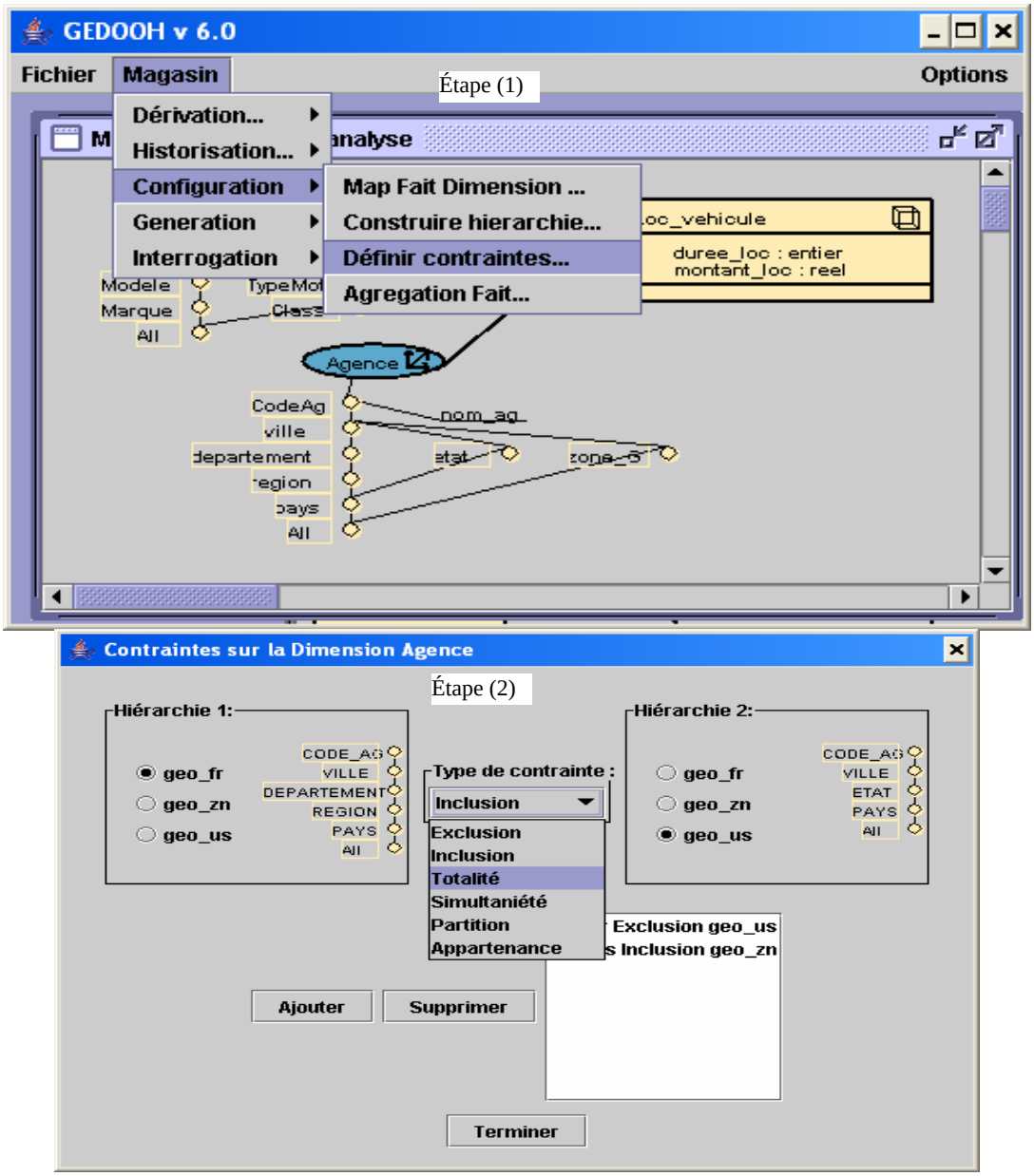


Figure V.13 *: Définition des contraintes intra-dimensions.*

4.6.2. Contraintes inter-dimensions

Il s'agit de contraintes portant sur les hiérarchies de dimensions distinctes reliées à un même fait. Nous rappelons que nous avons défini cinq contraintes à ce niveau : *exclusion, inclusion, partition, simultanéité et totalité.*

Ces contraintes sont définies de la même manière que les contraintes intra, à la différence que le concepteur sélectionne au début deux dimensions à la place d'une seule. L'interface graphique affiche automatiquement deux listes de hiérarchies correspondantes à chaque dimension (Figure V.14). Le concepteur peut alors choisir une hiérarchie dans chaque liste et définir le type de contrainte à appliquer sur les deux hiérarchies choisies.

Exemple 6

Pour définir une contrainte d'exclusion entre la hiérarchie '*géo_fr*' de la dimension *Agence* et la hiérarchie '*clas_us*' de la dimension *Vehicule*, le concepteur sélectionne les deux dimensions citées dans la fenêtre du magasin et choisit l'opération '*Définir contraintes…*'. La fenêtre de création de contraintes affiche les listes de contraintes des deux dimensions. Sur la Figure V.14 le concepteur a sélectionné les hiérarchies '*geo_fr*' de la dimension *Agence* et '*clas_us*' de la dimension *Véhicule*. Il choisit la contrainte d'exclusion et valide l'opération afin d'ajouter cette contrainte à la liste des contraintes définies entre les deux dimensions.

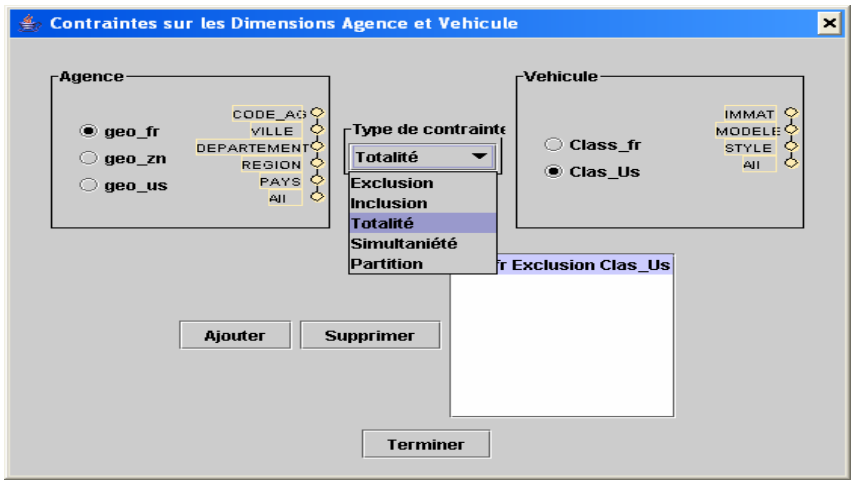


***Figure V.14** : Définition des contraintes inter-dimensions.*

4.7. Schéma de notre exemple de magasin de données

La réalisation des différentes étapes de notre démarche ascendante nous permet d'obtenir le schéma dimensionnel de la Figure V.15.

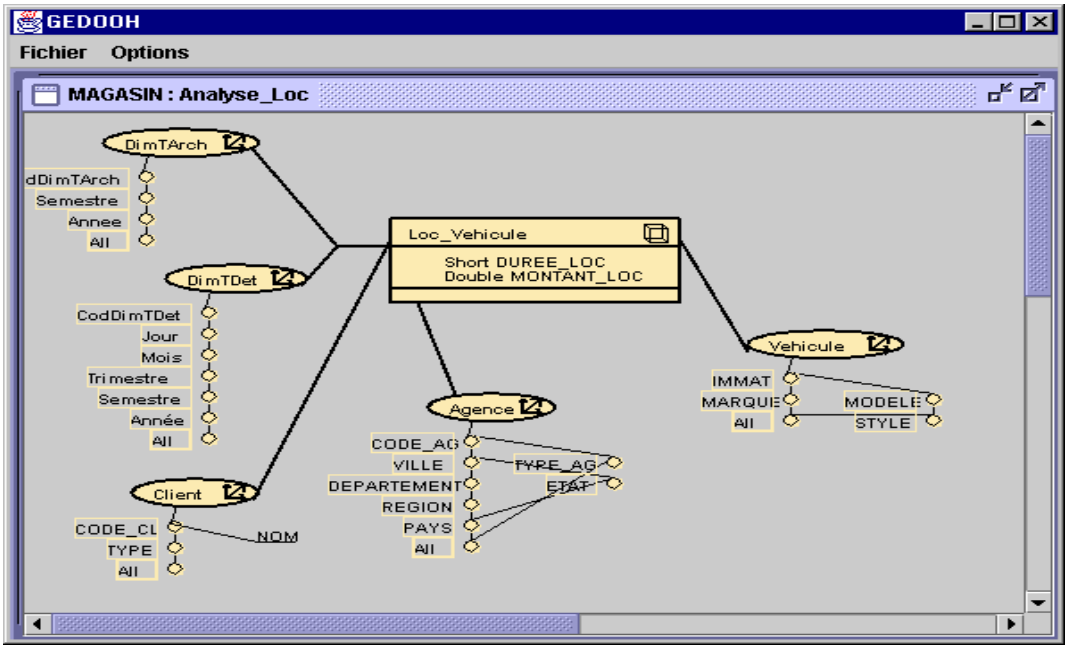


***Figure V.15** : Schéma dimensionnel du magasin Analyse_Loc*

5. Génération du magasin de données dimensionnelles

L'interface propose à l'administrateur un ensemble de fonctions de construction du schéma conceptuel de la base de données dimensionnelles. Les caractéristiques de ce schéma, stockées dans notre méta-modèle, seront utilisées par le générateur automatique afin d'implanter et de mettre à jour la base dimensionnelle.

Le générateur implante la base dans un SGBD relationnel (Oracle Version 8) et pour cela nous avons besoin de définir les règles de transformation assurant le passage entre le modèle dimensionnel (au niveau conceptuel) et le modèle relationnel (au niveau implantation).

Nous décrivons, dans ce qui suit, les règles de transformation dimensionnel - relationnel sur lesquelles reposent le générateur automatique. Puis, nous présentons les différents modules d'implantation physique du magasin de données dimensionnelles.

5.1. Phase logique

L'implantation du magasin de données dans un SGBD relationnel nécessite la transformation du schéma dimensionnel du magasin en un modèle relationnel. Nous présentons dans cette section les différentes règles de transformation des concepts dimensionnels en un modèle ROLAP.

5.1.1. Transformation des dimensions non temporelles

Une dimension (autre que la dimension Temps) est caractérisée par un identifiant et comporte un ensemble de paramètres et d'attributs faibles.

Règle 1. Une dimension (non temporelle) se transforme en une relation de même nom dont les attributs sont les paramètres de la dimension. La clé de cette relation est composée de l'identifiant de la dimension.

5.1.2. Transformation de la dimension temps

Selon notre modèle dimensionnel, deux hiérarchies sont définies dans la dimension Temps ; une hiérarchie qui décrit les données détaillées et une deuxième pour les données archivées. La transformation de cette dimension au niveau logique donne lieu à la définition de deux relations. La relation *T_Det* caractérise les mesures détaillées des faits et la relation *T_Arch* comporte les paramètres temporels des données archivées. Nous avons séparé les deux hiérarchies en deux relations afin d'harmoniser la création de la clé qui décrit dans chaque relation un niveau de granularité différent. Par exemple, le niveau jour pour la dimension détaillée et le niveau trimestre pour la dimension d'archive.

Règle 2. La hiérarchie temporelle *T_Det* se transforme en une relation de même nom dont les attributs sont les propriétés de la hiérarchie. La clé de cette table est définie par le système.

Règle 3. La hiérarchie d'archives *T_Arch* se transforme en une relation de même nom dont les attributs sont les propriétés d'archives. La clé de cette table est définie par le système.

5.1.3. Transformation des faits

Dans notre magasin, un fait comporte des données détaillées et/ou des données archivées. L'introduction de la hiérarchie temporelle détaillée et/ou de la hiérarchie d'archives dans le modèle dimensionnel nécessite un traitement spécifique qui permet de conserver la cohérence et la fiabilité des mesures calculées. Ainsi, nous proposons de transformer le fait en deux relations (tables) comportant, respectivement, les données historisées si le fait est dimensionné par *T_Det*, et les données archivées si *T_Arch* est une hiérarchie qui caractérise les mesures du fait.

Règle 4. Un fait détaillé (dimensionnée par *T_Det*) se traduit par une relation ayant le même nom précédé par «FH_», les mesures du fait comme attributs et comme clé la concaténation des clés des dimensions du fait.

La clé de cette table comporte, donc, la clé de la relation temporelle détaillée permettant de récupérer à chaque instant la date d'une mesure détaillée.

Si ce fait est archivé alors nous définissons une deuxième relation qui sera connectée à la relation temporelle d'archive *T_Arch* et aux autres dimensions.

Règle 5. Un fait archivé (dimensionné par *T_Arch*) se traduit par deux relations ; la première est représentée ci-dessus, la deuxième a le même nom que le fait précédé par «FA_», les mesures du fait comme attributs et comme clé la concaténation des clés des dimensions du fait. La clé de la deuxième table de fait ne comporte que la clé de la dimension d'archives permettant de récupérer les mesures agrégées durant la période d'archives.

Nous avons choisi de transformer notre modèle dimensionnel en un modèle logique ROLAP, en considérant que les SGBDs relationnels ont atteint une bonne maturité permettant d'optimiser le stockage et l'interrogation des données dimensionnelles. Ce choix n'exclut pas la possibilité de transformer notre modèle conceptuel en un modèle MOLAP ou OOLAP.

Dans la section suivante, nous décrivons la phase d'implantation du schéma ROLAP obtenu dans le SGBD relationnel ORACLE 9i.

5.2. Phase Physique

Après avoir défini le schéma conceptuel des données dimensionnelles du magasin et les règles de passage à partir de ce schéma vers un schéma relationnel, nous nous intéressons dans cette section à la génération physique de la base de données dimensionnelles. En effet, la génération automatique est réalisée en trois étapes :

- la première permet d'extraire les méta-données, décrivant le schéma du magasin à partir du référentiel des méta-données et d'implanter par la suite ce schéma dans la base de données relationnelles ;
- la deuxième étape est consacrée à l'initialisation du magasin à partir de l'entrepôt. C'est la première extraction qui va permettre d'alimenter le magasin ;
- enfin, l'étape de rafraîchissement permet de charger les données récentes à partir de l'entrepôt. L'entrepôt historisé permet de stocker l'évolution des données des sources opérationnelles. Cette évolution est propagée au niveau des magasins lors des rafraîchissements.

5.2.1. Création des schémas des magasins

La définition graphique du schéma dimensionnel basée sur notre démarche ascendante permet de définir toutes les caractéristiques nécessaires à l'implantation et à la maintenance de la base de données dimensionnelles. Ces caractéristiques sont stockées dans le référentiel de méta-données orientées objet.

A partir de cette description, un module de création se charge de générer les scripts de création du schéma de la base de données dimensionnelles représentant le magasin. Le module lance en même temps les commandes de création de ce schéma.

Une fois que le magasin est implanté dans la base, une première extraction des données de l'entrepôt est générée automatiquement. Cette extraction permet de peupler le magasin et de remplir les tables des faits et des dimensions.

5.2.2. Initialisation et rafraîchissement des magasins

Notre entrepôt de données permet de garder l'historique des évolutions des données des sources opérationnelles sous forme détaillées et/ou résumées. Les données historisées de l'entrepôt ne sont pas mises à jour mais rafraîchies d'une manière périodique afin d'insérer les informations récentes. Les magasins dimensionnels que nous avons conçus au niveau de notre outil présentent des extraits de l'entrepôt et sont à leur tour rafraîchis afin d'insérer les données récentes et de recalculer les mesures d'activité.

Le rafraîchissement est réalisé en insérant les nouveaux n-uplets des versions courantes de l'entrepôt dans les tables des faits et des dimensions et en modifiant les dates de fin de transaction pour les anciens n-uplets du magasin. En ce qui concerne les données agrégées, l'insertion de nouveaux n-uplets dans l'entrepôt nécessite la mise à jour de ces agrégats. Si, par exemple, nous insérons un nouveau n-uplet dans l'entrepôt correspondant au montant des location du magasin X pour le 29-sept-2004, ce montant doit être intégré dans le calcul du montant des locations du mois de septembre de l'année 2004 au niveau du magasin.

5.3. Bilan

Pour créer notre base dimensionnelle contrainte, nous avons proposé, tout d'abord, une interface graphique dans GMAG qui se caractérise par un fonctionnement incrémental, interactif, uniforme et flexible. Cette interface est implantée à l'aide du langage Java (jdk 1.4) assurant la portabilité de notre outil. La transformation logique du schéma conceptuel du magasin est également réalisée par un programme Java en se basant sur le méta-modèle objet du magasin. Le module de génération lancé à partir du module de l'interface réalise l'implantation physique des tables relationnelles du magasin dans un SGBD relationnel (Oracle, ou Access)

Actuellement, l'outil GMAG comprend environ 5000 lignes de code Java (jdk 1.4) pour l'interface et les modules de génération du magasin.

6. Conclusion

Dans ce chapitre, nous avons proposé un outil d'aide à la conception de magasins de données dimensionnelles à contraintes. Cet outil valide la démarche ascendante de notre méthode. Afin d'aider le concepteur à définir une base de données dimensionnelles qui répond aux besoins décisionnels, nous avons proposé une méthode mixte de conception de

schéma dimensionnel. Cette méthode est basée sur un modèle dimensionnel contraint joint par des formalismes graphiques. Elle comporte deux démarches : descendante basée sur les besoins des décideurs et ascendante partant de l'entrepôt de données pour concevoir le schéma dimensionnel contraint. L'outil présente l'avantage de proposer une définition graphique et incrémentale reposant sur une démarche précise (ordonnancement des étapes). En particulier, nous avons proposé :

- ***un référentiel de méta-données*** décrit par un diagramme de classe UML. Ce référentiel comporte les méta-données qui décrivent notre schéma dimensionnel contraint. Il est instancié par notre interface graphique lors de la définition des éléments du schéma dimensionnel conçu par le concepteur. Ce référentiel est consulté par le module de génération physique du magasin afin de réaliser la transformation du schéma conceptuel en un schéma logique et d'implanter les structures de ce schéma ;
- ***un langage de définition graphique*** basé sur une interface qui assiste l'administrateur dans la phase de conception du magasin de données dimensionnelles. Cette conception est basée sur un modèle dimensionnel en constellation. La démarche de conception ascendante implantée dans notre interface est composée de cinq étapes : la définition des faits ou centres d'analyse, la construction des dimensions pour chaque fait, l'organisation des hiérarchies dans les dimensions et enfin la définition des contraintes sémantiques assurant la cohérence des données du modèle ;
- ***Un générateur automatique*** qui réalise l'implantation des structures physiques des magasins de données dimensionnels dans un SGBD hôte. La génération est réalisée en deux phases logique et physique. Durant la phase logique nous transformons le schéma dimensionnel du magasin en schéma logique suivant un modèle ROLAP. La phase physique réalise l'implantation du magasin en trois étapes : création des schémas des magasins, peuplement de ses structures à partir de l'entrepôt et enfin une étape de rafraîchissement des magasins afin de charger les informations récentes à partir de l'entrepôt.

L'implantation physique obtenue respecte les contraintes exprimées dans notre schéma conceptuel dimensionnel. Ainsi, nous avons validé nos propositions par l'application complète de la démarche proposée, de la description graphique des schémas conceptuels jusqu'à l'implantation dans un SGBD hôte des bases dimensionnelles contraintes.

BILAN & PERSPECTIVES

Bilan

Nos travaux de recherche se situent dans le cadre des systèmes d'aide à la décision. Ces systèmes se basent généralement sur une approche OLAP afin de faciliter l'interrogation et l'analyse des données. Cette approche adopte la modélisation dimensionnelle organisant les données d'une manière adaptée aux analyses (Kimball et al, 2002).

Dans le cadre de cette thèse, nous proposons un modèle de données dimensionnel contraint, un langage de manipulation de données adapté ainsi qu'une méthode de conception de schéma dimensionnel.

- **Modèle dimensionnel contraint**. Le modèle conceptuel que nous avons défini, apporte des solutions nouvelles répondant aux exigences de fiabilité et de cohérence des bases de données dimensionnelles (Hurtado et al, 2002). Ce modèle conceptuel permet de faire abstraction de toute spécificité d'ordre technique. Il constitue une généralisation des modèles dimensionnels habituellement proposés (constellation de faits et dimensions munies de hiérarchies multiples). L'autre intérêt de notre modèle réside dans sa capacité à exprimer les contraintes structurelles et sémantiques. Cette problématique est peu traitée dans la littérature (Carpani et al, 2001). Les contraintes structurelles assurent la validité du schéma dimensionnel. Notamment, elles permettent de concevoir des hiérarchies valides au sein des dimensions afin de réaliser correctement les opérations d'agrégation. Les différents types de contraintes sémantiques (exclusion, inclusion, simultanéité, totalité, partition), exprimées entre les hiérarchies d'une ou de plusieurs dimensions, permettent de modéliser les règles de gestion du monde réel. L'intégration de ces contraintes permet de restituer des données cohérentes et fiables lors de l'interrogation de données dimensionnelles. De plus, notre modèle offre une vision détaillée ou archivée des données décisionnelles. Cette problématique est résolue par la proposition d'une dimension temps à hiérarchies multiples.
- **Manipulation des données dimensionnelles**. Dans le contexte OLAP, nous avons proposé un langage algébrique de manipulation de données dimensionnelles contraintes. Ce langage est basé sur une visualisation tabulaire et interactive des données dimensionnelles facilitant l'analyse pour un décideur non informaticien. Notamment, nous avons étendu les opérateurs dimensionnels généralement proposés dans la littérature (forage, rotation, …) afin de leur conférer une propriété supplémentaire offrant la possibilité de maintenir ou d'étendre les analyses en fonction des contraintes sémantiques existantes entre les hiérarchies. Cette proposition contribue à améliorer la cohérence des analyses (Ghozzi et al, 2003c). A notre connaissance, l'intégration des contraintes lors de la manipulation des données dimensionnelles n'a fait l'objet d'aucun travail de recherche.
- **Méthode de conception des données dimensionnelles contraintes**. Afin d'aider le concepteur à définir un schéma dimensionnel fiable et complet, nous avons proposé une méthode de conception de schéma dimensionnel intégrant l'expression des contraintes sémantiques. L'avantage de notre méthode réside dans l'adoption d'une approche mixte permettant de définir le schéma dimensionnel en se basant sur les besoins des décideurs (démarche descendante) et tenant compte des données de l'entrepôt (démarche ascendante) (Trujillo et al, 2003). La démarche descendante réalise la collecte (interviews, questionnaires, études des rapports existants), la

spécification (matrice des besoins, langage de définition des contraintes LCD) et la formalisation des besoins des décideurs sous forme de schéma dimensionnel contraint. En parallèle, la démarche ascendante permet de collecter les données sources et de construire les composantes du schéma dimensionnel (fait, dimension, hiérarchies) en suivant un ensemble d'étapes ordonnées. Ces étapes peuvent être réalisées d'une manière semi-automatique en se basant sur les algorithmes de définition des dimensions et de hiérarchies. Une phase de confrontation permet d'intégrer les résultats des deux démarches pour obtenir un schéma dimensionnel contraint intégrant à la fois les besoins des décideurs et les données sources.

- **Outil d'aide à la conception de magasin dimensionnel contraint**. Afin de valider nos travaux, nous avons réalisé un outil d'aide à la conception de schéma dimensionnel contraint à partir du schéma d'un entrepôt de données historisées. Notre outil assiste le concepteur dans la définition du schéma dimensionnel de manière graphique et incrémentale. Pour ce faire, nous avons proposé un langage de conception graphique basé sur les étapes de la démarche ascendante. Avec notre outil, nous déchargeons le concepteur de la phase de génération du magasin. En effet, grâce au référentiel de méta-données qui stocke les caractéristiques du schéma dimensionnel contraint du magasin, un générateur réalise l'implantation automatique du magasin dans un SGBD hôte.

Nos travaux de thèse ont permis de proposer une méthode complète de conception de bases de données dimensionnelles. Cette méthode comporte un modèle dimensionnel contraint (concepts et formalismes graphiques), un langage de manipulation, une démarche de conception de schémas dimensionnels et un outil d'aide à la conception.

Perspectives

Les perspectives que nous envisageons de conduire sont les suivantes :

- **Une extension de notre modèle dimensionnel contraint.** Notre proposition permet de modéliser les besoins des décideurs sous forme de schéma dimensionnel contraint. Or, ces besoins peuvent évoluer dans le temps (nouvel indicateur ou paramètre d'analyse, nouvel axe d'analyse, …). La prise en compte de ces nouveaux besoins peut s'effectuer au travers de la gestion de l'évolution du schéma dimensionnel. Plus précisément, cette étude pourrait se centrer sur la gestion de l'historique des mesures et des paramètres voire la gestion de l'historique des faits et des dimensions (Bellahsène, 2002). En outre, l'accès à l'information décisionnelle est vital pour l'avenir de l'entreprise d'où le besoin de sécuriser l'accès aux données dimensionnelles. Ceci nécessite la gestion des droits d'accès des utilisateurs aux différentes structures dimensionnelles (fait, dimension, hiérarchie). Cette perspective a fait l'objet d'une première proposition (Sallami, 2004) gérant les droits d'accès des décideurs aux différentes composantes du schéma dimensionnel.
- **Une extension de notre langage d'interrogation des données dimensionnelles**. Nous avons proposé un langage algébrique d'interrogation de données dimensionnelles comportant un ensemble d'opérateurs unaires qui permettent d'afficher et de modifier une table dimensionnelle pour créer une nouvelle. Or, ces opérateurs ne permettent pas de comparer deux tables dimensionnelles visualisées.

Une extension possible de ces opérateurs consiste à proposer des opérateurs binaires permettant de fusionner deux tables dimensionnelles (union, intersection, …) afin de faciliter la corrélation entre les analyses. En outre, notre langage algébrique basé sur le concept de table dimensionnelle, ne propose pas l'interrogation graphique directe des schémas dimensionnels. Or, l'interrogation des données décisionnelles basée sur des schémas simples et faciles à manipuler peut simplifier la tâche des décideurs (non informaticiens). Pour répondre à ce besoin, nous envisageons de proposer un langage de manipulation graphique des schémas dimensionnels en constellation.

- **Une extension de notre méthode de conception**. Avec la croissance du Web et de l'Internet, les documents représentent une source d'information importante que l'entreprise peut intégrer dans son système décisionnel. Dans ce contexte, nous avons assisté à l'apparition du standard XML (eXtensible Markup Langage) permettant d'unifier et de structurer la représentation des documents. Une extension possible de notre méthode de conception de schéma dimensionnel concerne la prise en compte des documents de type XML. La problématique dans ce contexte réside dans l'hétérogénéité des structures des documents traités difficiles à intégrer dans une base de données possédant une structure unique. Une des solutions possibles est de définir des familles de structure logique générique permettant de décrire tous les documents (Khrouf et al, 2003). Nous envisageons d'étudier des collections de documents spécialisées notamment celles qui comportent des données factuelles (données économiques, statistiques,…). Le choix d'une collection spécialisée permet de la représenter par une seule structure générique que nous transformons en un diagramme de classes UML (Jensen et al, 2003). Nous proposons d'intégrer le diagramme UML résultant de cette transformation dans le schéma de l'entrepôt et d'y appliquer notre méthode de conception pour dériver le schéma dimensionnel. Nous notons que l'intégration des structures des documents XML en une structure générique nécessite l'analyse sémantique des documents (Jouve et al, 2003).

Ces perspectives étendent nos propositions aussi bien aux niveaux du modèle et du langage d'interrogation que de la méthode de conception. D'autres perspectives peuvent être envisageables à long terme pour nos recherches telle que la gestion des données incertaines au niveau du modèle, du langage d'interrogation et de la méthode permettant de tenir compte des imprécisions du monde réel.

BIBLIOGRAPHIE

A

(Abello et al, 2002) Abelló A., Samos J, Saltor F. *"YAM2 (Yet Another Multidimensional Model): An Extension of UML"*. Dans 5th Iberoamerican Workshop Requirements Engineering and Software Environments (IDEAS'02), La Habana, Cuba, p. 172-181, avril 2002.

(Abello et al, 2003) Abelló A., Samos J., and Saltor F. *"Implementing Operations to Navigate Semantic Star Schemas"*. Dans 6th International Workshop on Data Warehousing and OLAP (DOLAP'03). New Orleans, USA, ACM, p. 56-62, novembre 2003.

(Agrawal et al, 1997) Agrawal R., Gupta A., Sarrawagi A., *"Modeling Multidimensional Databases"*. Dans 13th International Conference on Data Engineering (ICDE'97), Birmingham, UK, p. 232- 243, avril 1997.

B

(Baralis et al, 1997) Baralis E., Paraboschi S., Teniente E., *"Materialized Views Selection in a Multidimensional Database"*. Dans 23rd International Conference on Very Large Data Bases (VLDB'97), 25-29 août 1997, p. 156-165, Athènes, Grecs.

(Baril et al, 2003) Baril X., Bellahsène Z., *"Selection of Materialized Views: A Cost-Based Approach"*. Lecture Notes in Computer Science, Springer-Verlag Heidelberg ISSN: 0302-9743 Computer Science, Vol. 2681, p. 665-680, 2003.

(Bellahsène, 2002) Bellahsene Z., *"Schema Evolution in Data Warehouses"*. Knowledge and Information Systems, Springer-Verlag London Ltd, Vol. 4, N.3, p. 283-304, juillet 2002.

(Bellahsène et al, 1999) Bellahsène Z., Hacid M. S. "*A Knowledge Based Approach for Modeling and Querying Multidimensional Databases*", in Proc. of 10th International Conference on Database and Expert Systems Applications (DEXA'99), Lectures Notes in Computer Science, Springer Verlag, Florence, September 1999.

(Bellatreche, 2000) Bellatreche L., "Utilisation *de la Fragmentation, des Vues Matérialisées et des Index dans la Conception d'un Entrepôt de Données"*. Thèse de doctorat en informatique, Univ. Clermont Ferrand (collaboration) soutenue le 18 décembre 2000.

(Blashka et al, 1998) Blashka M, Sapia C., Höfling G., Dinter B., *"Finding your way through multidimensional data model"*. Dans 9th International Workshop on Database and Expert Systems Applications (DEXA'98), Viennes, Autriche, IEEE Computer Society Press, p. 198-203, août 1998.

(Bret et al, 1999) Bret F., Teste O., *"Construction Graphique d'Entrepôts et de Magasins de Données"*. Dans 17ème Congrès INFormatique des ORganisations et Systèmes d'Information et de Décision (INFORSID'99), La Garde, France, p.165-184, Juin 1999.

(Bruckner et al, 2001) Bruckner R. M.; List, B., Schiefer, J., Tjoa, A M. *"Modeling Temporal Consistency in Data Warehouses"*. Dans 12th International Workshop on Database and Expert Systems Applications (DEXA'01), first International Workshop on Knowledge Extraction for Enterprise Services (KEES'01), Munich, Allemagne, IEEE Computer Society Press, p. 901-905, septembre 2001.

(Bukhres et al, 1993) Bukhres O.A., Elmagarmid A.K., *"Object-Oriented Multidatabase Systems - A solution for Advanced Applications"*, Prentice Hall, ISBN 0-13-103813-2, 1993.

(Buzydlowski et al, 1998) Buzydlowski J. W, Song II-Yeol, Hassel L., *"A framework for object oriented online Analytical processing"*. Dans 1st International Workshop on Data Warehousing and OLAP (DOLAP'98), Bethesda, Maryland, USA, ACM, p 10-15, novembre 1998.

C

(Cabibbo et al, 1998) Cabibbo L., Torlone R., *"A Logical Approach to Multidimensional Databases"*. Dans 6th International Conference on Extending Database Technology (EDBT'98), Valencia, Espagne, Springer, Lecture Notes in Computer Science 1377, p. 183-197, mars 1998.

(Cabibbo et al, 2000) Cabibbo L., R. Torlone. *"The Design and Development of a Logical OLAP System"*. Dans 2nd International Conference of Data Warehousing and Knowledge Discovery (DaWaK'00), London, UK, Springer, Lecture Notes in Computer Science 1874, p. 1-10, Septembre 2000.

(Cavero et al, 2001) Cavero J., Piattini M., Marcos E., *"MIDEA : A multidimensional Data Warehouse Methodology"*. Dans 3rd international Conference on Entreprise Information Systems (ICEIS'01), Setubal, Portugal, p. 138-144, juillet 2001.

(Carneiro et al, 2002) Carneiro L., Brayner A., *"X-META : A Methodology for Data Warehouse Design with Metadata Management"*. Dans 4th International Workshop on Design and Management of Data Warehouses (DMDW'02), Toronto, Canada, p. 13-22, mai 2002.

(Carpani et al, 2001) Carpani F., Ruggia R., *"An Integrity Constraints Language for a Conceptual Multidimensional Data Model"*. Dans 13th International Conference on Software Engineering & Knowledge Engineering (SEKE'01), Argentina, 2001.

(Chaudhuri et al, 1997) Chaudhuri S., Dayal U., *"An overview of data warehousing and OLAP technology"*, ACM SIGMOD record, Vol. 26, N. 1, p517-526, mars 1997.

(Choong et al, 2003) Choong Y. W., Laurent D. and Marcel P. *"Computing appropriate representations for multidimensional data"*. Data & Knowledge Engineering, Vol. 45, N. 2, p. 181-203, mai 2003.

(Codd, 1970) Codd E. F. A *"Relational Model of Data for Large Shared Data Banks"*. Communications of the ACM (CACM), Vol. 13, N. 6, p. 377-387, 1970.

(Codd et al, 1993) Codd E. F., Codd S.B., Salley C.T., *"Providing OLAP (On Line Analytical Processing) to Users-Analysts: An IT Mondate"*, Rapport technique, E.F. Codd and Associates, 1993.

D, E, F

(De Miguel et al, 2000) De Miguel A., Cavero J. M., Canela J., Sanchez A. *"IDEA-DWCASE: Modeling Multidimensional Databases"*. Dans 7th Conference on Extending Database Technology (EDBT'00) - Featured Software Demonstrations, Konstanz, Allemagne, Mars 2000.

(Doucet et al, 1996) Doucet A., Gançarski S., Jomier G., Monties S. *"Integrity Constraints and Versions"*. Dans 6th International Workshop on Foundations of Models and Languages for Data and Objects "Integrity in Databases" (FMLDO'96). Dagstuhl Castle, Allemagne, p. 25-39, septembre 1996.

(Ferrat, 1983) Lounas Ferrat, *"Expression et contrôle de l'intégrité sémantique dans les bases de données relationnelles Projet Microbe"*, Thèse de doctorat, Université scientifique et médicale de Grenoble et INP de Grenoble, 1983.

G

(Gardarin, 1999) Gardarin G., *"Bases de données",* Eyrolles, Janvier 1999.

(Ghozzi, 2002) Ghozzi F, "*Modèle multidimensionnel temporel à contraintes"*. Dans : *Colloque des Doctorants de l'Ecole Doctorale Informatique et Télécommunication (EDIT'02)*, *Toulouse*, p. 3-7, mars 2002.

(Ghozzi, 2003a) Ghozzi F., *"Matérialisation des vues dans un modèle multidimensionnel contraint"*. Dans : *Congres de l'INFormatique des Organisations et Systemes d'Information et de Decision (INFORSID'03)*, INFORSID-édition (Eds.), Nancy, France, p. 351-368, juin 2003(Prix meilleur article "Catégorie Jeune Chercheur").

(Ghozzi, 2003b) Ghozzi F., *"Matérialisation des vues dans un modèle multidimensionnel contraint"*. Dans Revue ISI - RSTI Systèmes d'information avancés, Hermes -Lavoisier, Vol. 8, N. 4, p. 9-34, novembre 2003 (Version étendue Inforsid'03).

(Ghozzi et al, 2003a) Ghozzi F., Ravat F., Teste O., Zurfluh G., "*Modèle Dimensionnel à Contraintes"*. Dans Revue des Sciences et Technologies de l'Information, Série RIA-ECA, Hermes –Lavoisier, Vol. 17, N. 1-2-3, p.43-56, 2003.

(Ghozzi et al, 2003b) Ghozzi F, Ravat F., Teste O., Zurfluh G., "*Contraintes pour modèle et langage multidimensionnels"*. Dans 19ème Journées Bases de Données Avancées – (BDA'03), Lyon, France, Claude Chrisment (Eds.), p. 383-402, octobre 2003.

(Ghozzi et al, 2003c) Ghozzi F, Ravat F., Teste O., Zurfluh G., *"Constraints and multidimensional databases"*. Dans 5th International Conference on Enterprise Information Systems (ICEIS'03), Angers, France, p. 104-111, avril 2003.

(Ghozzi et al, 2004) Ghozzi F, Ravat F., Teste O., Zurfluh G., "*Contraintes pour modèle et langage multidimensionnels"*. Dans RSTI-ISI : Fouille, transactions, évaluation dans les bases de données, Hermes –Lavoisier, Vol. 9, N. 1, p.9-34, 2004.

(Golfarelli et al, 1998) Golfarelli M., Rizzi S.,*"A methodological Framework for Data Warehouse Design"*, Dans 1st International Workshop on Data Warehousing and OLAP (DOLAP'98), Bethesda, Maryland, USA, p. 3-9, novembre 1998.

(Golfarelli et al, 2002) Golfarelli M., Rizzi S., Saltarelli E. *"WAND: A CASE Tool for Workload-Based Design of a Data Mart"*. Dans 10th National Convention on Systems Evolution for Data Bases (SEBD'02), Portoferraio - Island of Elba, Italie, ed. Paul Ciaccia, the Faustus Rabitti, Hard Giovanni, p. 422-426, juin 2002.

(Gray et al., 1996) Gray J., Bosworth A., Layman A., and Pirahesh H., *"Data cube: a relational aggregation operator generalizing group-by, cross-tabs and subtotals"*. 12th International Conference on Data Engineering (ICDE'96), New Orleans, Louisiana, USA, p. 152-159, mars 1996.

(Gupta et al, 1997) Gupta H., Harinarayan V, Rajaraman A., *"Index selection for OLAP"*. Dans 13th International Conference on Data Engineering (ICDE'97), Birmingham, U.K, p. 208-219, avril 1997.

(Gupta et al, 1999) Gupta H., Mumick I., S., *"Selection of view to materialize under a maintenance cost constraint"*. Dans 7th International Conference Database Theory (ICDT '99), Jerusalem, Israel, p. 453-470, janvier 1999.

(Gyssen et al, 1997) Gyssen M., Lakshmanan L. V. S., *"A Foundation for Multi-Dimensional Databases"*. Dans 23rd International Conference on Very Large Data Bases (VLDB'97), Athènes, Grecs, p. 106-115, août 1997.

H

(Hahn et al, 2000) Hahn K., Sapia C., Blaschka M., *"Automatically Generating OLAP Schemata from Conceptual Graphical Models"*. Dans 3rd International Workshop on Data Warehousing and OLAP (DOLAP'00, in connection with CIKM), Washington, USA, novembre 2000, p. 9-16.

(Harinarayan et al, 1996) Harinarayan V., Rajaraman A., Ullman J. *"Implementing Data Cubes Efficiently"*. Dans ACM – SIGMOD International Conference on Management of Data, Montreal, Canada, SIGMOD record, Vol. 25, N. 2, p. 205-216, juin1996.

(Harkins, 2003) Harkins S. "Améliorez l'intégrité de vos données par une conception optimale", URL:http://www.zdnet.fr/builder/architecture/base_de_donnees/0,39020907,-21351871,00.htm, 26 mai 2003.

(Hümmer et al, 2002) Hümmer W., Lehner W., Bauer A., Schlesinger L., *"*A Decathlon in Multidimensional Modeling: Open Issues and Some Solutions*"*. Dans 4th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2002), Aix-en-Provence, France, p. 275-285, September, 2002.

(Hurtado et al, 1999) Hurtado C A., Mendelzon A. O., VaismanA.A. *"Maintaining Data Cube under Dimension Update"*. Dans 15th International Conference on Data Engineering (ICDE'99), Sydney, Australie, p. 346-355, mars 1999.

(Hurtado et al, 2002) Hurtado C.A., Mendelzon A.O., *"OLAP Dimension Constraints"*. Dans 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'02), Madison, USA, p. 169-179, juin 2002.

I, J, K

(Inmon, 1996) Inmon W. H., *"Building the Data Warehouse"*. John Wiley & Sons, deuxième édition, ISBN 04771-14161-5, 1996.

(Jarke et al, 2003) Jarke M., Lenzerini M., Vassiliou Y., Vassiliadis P., *"Fundamentals of Data Warehouses"*. Springer-Verlag, deuxième édition, ISBN 3-540-42089-4, 2003.

(Jensen et al, 2003) Jensen M. R., Møller T. H., Pedersen T. B., *"Converting XML DTDs to UML diagrams for conceptual data integration"*. Dans Data & Knowledge Engineering, Elsevier Science B.V. edition, Vol 44, N 3, p. 323–346, mars 2003.

(Jouve et al, 2003) Jouve D., Amghar Y., Chabbat B., Pinon J. -M., "Conceptual framework for document semantic modelling: an application to document and knowledge management in the legal domain". Dans Data & Knowledge Engineering, Elsevier B.V., Volume 46, Issue 3, Pages 345-375, September 2003

(Khrouf et al, 2003) Khrouf K., Ravat R., Soulé-Dupuy C., "*Comparaison et fusion de structures logiques de documents semi-structurés*". Dans : *Ingénierie des Systèmes d'Information (ISI)*, Hermès, V. 8, N. 5-6, p. 127-151, 2003.

(Kimball et al, 2002) Kimball R., Ross M., *"The Data Warehouse Toolkit"*, Wiley, New York, deuxième édition, 2002.

(Kotidis et al, 2001) Kotidis Y., Roussopoulos N., *"A case for dynamic view management"*, Database Systems Journal, Vol. 26, N. 4, p. 388-423, 2001.

L

(Lapujade, 1997) Lapujade A. "Définition d'un méta-modèle et préservation de son intégrité dans le méta-atelier de conception de systèmes de formation MATIF", Thèse de doctorat de l'Université Toulouse I, 1997.

(Le Parc, 1997) Le Parc A., "Une algèbre et un langage graphique pour les bases de données objet intégrant le concept de version", Thèse de l'Université Paul Sabatier - Toulouse III, Décembre 1997.

(Lechtenbörger et al, 2003) J. Lechtenbörger, G. Vossen *"Multidimensional normal forms for data warehouse design"*. Dans Revue Information Systems, Vol. 28, N. 5, p. 415-434, juillet 2003.

(Lehner, 1998) Lehner W., "Modeling Large Scale OLAP Scenarios". Dans 6th International Conference on Extending Database Technology (EDBT'98), Valence, Espagne, p. 153-167, mars 1998.

(Li et al, 1996) C. Li, X.S.Wang, *"A Data Model for supporting on-line analytical processing"*. Dans 5th International Conference on Information Knowledge Management (CIKM'96), Rockville, Maryland, USA, p. 81- 88, novembre 1996.

(Lim et al, 2004) Lim Y., Kim M., *"A Bitmap Index for Multidimensional Data Cubes"*. Dans 15th International Conference on Database and Expert Systems Applications (DEXA'04), Zaragoza, Espagne, Septembre 2004.

(List et al, 2002) List B., Bruckner R., Machaczek K., Sciefer J., *"A Comparison of Data warehouse development Methodologies case study of the process Warehouse"*. Dans 13th International Conference on Database and Expert Systems Applications (DEXA'02), Aix-en-Provence, France, LNCS 2453, p 203-215, Septembre 2002.

(Lujan et al, 2004) Luján-Mora S., Trujillo J., Vassiliadis P. *"Advantages of UML for Multidimensional Modeling"*. Dans 6th International Conference on Enterprise Information Systems (ICEIS'04), Porto, Portugal, p. 298-305, avril, 2004.

M

(Marcel, 1998) Marcel P. *"Manipulation de Données Multidimensionnelles et Langages de Règles"*, Thèse de Doctorat de l'Institut des Sciences Appliquées de Lyon, 1998.

(Mendelzon et al, 2000) Mendelzon A. O., Vaisman A.A., *"Temporal queries in OLAP"*. Dans 26th international Conference on Very Large Data Bases (VLDB 2000), Cairo, Egypte, p. 242-253, Septembre 2000.

(Mendelzon et al, 2003) Mendelzon A. O., Vaisman A.A., *"Time in Multidimensional Databases"*. Dans "Multidimensional Databases: Problems and Solutions". Idea Group Inc., IGP/INFOSCI/IRM Press, Hershey, PA - USA, p 166-199, juin 2003.

(Mkaouar et al, 2003) Mkaour M., Bouaziz R., "*A support toolset for the development in a temporal database environment*". Dans International Conference in computer systems and applications (AICCSA'03), Tunis, Tunisie, Juillet 2003.

(Moody et al, 2000) Moody D. L., Kortink M. A.R. *"From Enterprise Models to Dimensional Models:A Methodology for Data Warehouse and Data Mart Design"*. Dans 2nd

International Workshop on Design and Management of Data Warehouses (DMDW'2000) Stockholm, Suède, papier 5, juin 2000.

(Muller, 2000) Muller P-A., *"Modélisation objet avec UML"*, ed. Eyrolles, ISBN 2212091222, Mars 2000.

N, O

(Nguyen et al, 2000) Nguyen T. B., TJOA A. M., Wagner R.R. *"An Object Oriented Multidimensional Data Model for OLAP"*. Dans 1st International Conference on Web-Age Information Management (WAIM'00), Shanghai, Chine, Springer LNCS 1846, 2000, p. 69-82, juin 2000.

P, Q

(Paraboschi et al, 2003) Paraboschi S., Sindoni G., Baralis E., Teniente E. *"Materialized Views in Multidimensional Databases"*. Dans "Multidimensional Databases: Problems and Solutions" Idea Group Inc., IGP/INFOSCI/IRM Press, Hershey, PA - USA, juin 2003.

(Pedersen et al, 1998) Pedersen T.B., Jensen C.S., *"Research Issues in Clinical Data Warehousing"*. Dans 10th International Conference on Scientific and Statistical Database Management (SSDBM'98), Capri, Italie, p.43-52, juillet 1998.

(Pedersen et al, 1999) Pedersen T.B., Jensen C.S. *"Multidimensional Data Modeling for Complex Data"*. Dans 15th International Conference on Data Engineering (ICDE'99), Sydney, Australie, p. 363-345, mars1999.

(Prat et al, 2002) Prat N., Akoka J., "*From UML to ROLAP multidimensional databases using a pivot model"*. Dans 18ème journées Bases de données avancées (BDA02), Evry , France, p. 171-195, octobre 2002.

R

(Ravat et al, 1999) Ravat F., Teste O., Zurfluh G. "*Towards the Data Warehouse Design"*. Dans International Conference on Information Knowledge Management (CIKM'99), Kansas City, Kansas, USA, p. 359-366, novembre 1999.

(Ravat et al, 2000a) Ravat F., Teste O., *"Object-Oriented Decision Support System"*. Dans 2nd International Conference on Enterprise Information Systems (ICEIS'00), Stafford, UK, eds. B. Sharp, J. Cordeiro, J. Filipe, ISBN 972-98050-1-6, p.79-84, juillet 2000.

(Ravat et al, 2000b) Ravat F., Teste O., *"An Object Data Warehousing Approach : a Web Site Repository"*. Dans 4th East-European Conference on Advances in Databases and

Information Systems (ADBIS-DASFAA'00), Prague, Czech Republic, p. 128-137, Septembre 2000.

(Ravat et al, 2001) Ravat F., Teste O., Zurfluh G., *"Modélisation dimensionnelle des systèmes décisionnels"*. Dans Revue extraction des connaissances et apprentissage (ECA), Vol. 1, N. 1-2, p. 201-212, 2001.

(Ravat et al, 2002) Ravat F., Teste O., Zurfluh G., *"Langage pour Bases Multidimensionnelles: OLAP-SQL"*. Dans Revue ISI-NIS, ISBN 2-7462-0579-3, Vol. 7, N. 3, p.11-38, 2002.

(Red, 1997) Red Brick Systems. *"Star schema processing for complex queries"*. White Paper, juillet 1997.

S

(Sallami, 2004) Sallami M., "*Bases de données Multidimensionnelles : Langage de contrôle de données",* DEA Informatique de l'Image et du Langage, Université Paul Sabatier, juin 2004.

(Samtani et al, 1998) Samtani S., Mohania M. K., Kumar V., Kambayashi Y., *"Recent Advances and Research Problems in Data Warehousing"*. Dans Workshops on Data Warehousing and Data Mining, Mobile Data Access, and Collaborative Work Support and Spatio-Temporal Data Management (ER '98), Singapore, Springer LNCS 1552, p. 81-92, novembre 1998.

(Sanchez et al, 1999) Sanchez A., Cavero J.M., De Miguel A. *"IDEA: A conceptual multidimensional data model and some methodological implications"*. Dans Congreso Internacional de Investigacin en Ciencias Computacionales (CIICC'99), Cancun, Mexico, p. 307-318, septembre1999.

(Sapia et al, 1999) C. Sapia, M. Blaschka, G. Höfling, B. *"Dinter : Extending the E/R Model for the Multidimensional Paradigm"*. Dans Workshops on Data Warehousing and Data Mining, Mobile Data Access, and Collaborative Work Support and Spatio-Temporal Data Management (ER '98) DWDM, Singapore, Springer LNCS 1552, p. 105-116, novembre 1998.

T

(Teste, 2000) Teste O., *"Modélisation et Manipulation d'Entrepôts de Données Complexes et Historisées "*. Thèse de l'Université Paul Sabatier - Toulouse III, Décembre 2000.

(Theodoratos et al, 1999) Theodoratos D., Bouzeghoub M., *"Data Currency Quality Factors in Data Warehouse Design"*. Dans International Workshop on Design and Management of Data Warehouses (DMDW'99), Heidelberg, Allemagne, juin1999.

(Torlone, 2003) Torlone R., *"Conceptual Multidimensional Models"*. Dans "Multidimensional Databases: Problems and Solutions" Idea Group Inc., IGP/INFOSCI/IRM Press, Hershey, PA - USA, p 69-90, juin 2003.

(Tsois et al, 2001) Tsois A., Karayannidis N., Sellis T., *"MAC: Conceptual Data Modeling for OLAP"*. Dans International Workshop on Design and Management of Data Warehouses (DMDW'2001) Interlaken, Switzerland, juin 2001.

(Trujillo et al, 2001) Trujillo J. C., Palomar M., Gómez J., Song:I *"Designing Data Warehouses with OO Conceptual Models"*. Dans IEEE Computer, Vol. 34, N.12, p. 66-75, 2001.

(Trujillo et al, 2002) Trujillo J. C., Luján-Mora S., Medina E.. *"The GOLD Model CASE Tool: an environment for designing OLAP applications"*. Dans 4th International Conference on Enterprise Information Systems (ICEIS'02), Ciudad Real, Espagne, p. 699-707, avril 2002.

(Trujillo et al, 2003) Trujillo J. C., Luján-Mora S., Song I., *"Applying UML for designing multidimensional databases and OLAP applications"*. Dans K. Siau (Ed.), Advanced Topics in Database Research, Vol. 2, Hershey, PA: Idea Group Publishing, p. 13-36, 2003.

(Tryfona et al, 1999) Tryfona N., Busborg F., Borch Christiansen J.G., *"StarER : A Conceptual Model for Data Warehouse Design"*. Dans 2nd International Workshop on Data Warehousing and OLAP DOLAP'99, Kansas city, Missouri, USA, p. 3-8, Novembre 1999.

U, V

(Ullman, 1996) Ullman. J. "Efficient implementation of data cubes via materialized views" Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96), p. 386-388, 1996.

(Valduriez, 1987) Valduriez P., *"Join indices"*. Dans ACM Transactions on Database Systems, Vol. 12, N. 2, p. 218-246, juin 1987.

(Vassiliadis, 1998) Vassiliadis P., *"Modelling Multidimensional Databases* Dans 10th International Conference on Scientific and Statistical Database Management (SSDBM'98), Capri, Italie, p. 53-62, juillet 1998.

(Vassiliadis et al, 1999) Vassiliadis P., Sellis T., "*A Survey of Logical Models for OLAP Databases*". SIGMOD Record, Vol 28, N. 4, p. 64-69, December 1999.

(Villacampa, 2002) Villacampa F. *"Olap: analyser les données de l'entreprise"*, Décision Micro, 05/08/2002 URL:http://www.01net.com/article/189225.html.

W, X, Y, Z

(Widom, 1995) Widom J., "*Research problems in data warehousing*", Dans International Conference on Information and Knowledge Management (CIKM95), Baltimore, Maryland, USA, p. 25-30, Novembre 1995.

(Wu et al, 2004) Wu K., Otoo E., Shoshani A., "*On the Performance of Bitmap Indices for High Cardinality Attributes*". Dans 30th International Conference on Very Large Data Bases (VLDB'04), Toronto, Canada, Septembre 2004.

(Yang et al, 2000) Yang J., Widom J., "*Temporel View Self-Maintenance in a warehousing Environment*", Dans 7th International Conference on Extending Database Technology (EBDT'00), Konstanz, Allemagne, pp. 395-412, Mars 2000.

ANNEXE : OUTILS INDUSTRIELS

Dans le cadre industriel, le marché des applications décisionnelles rassemble plusieurs types d'outils pouvant être classés en trois catégories suivant l'architecture suivante :

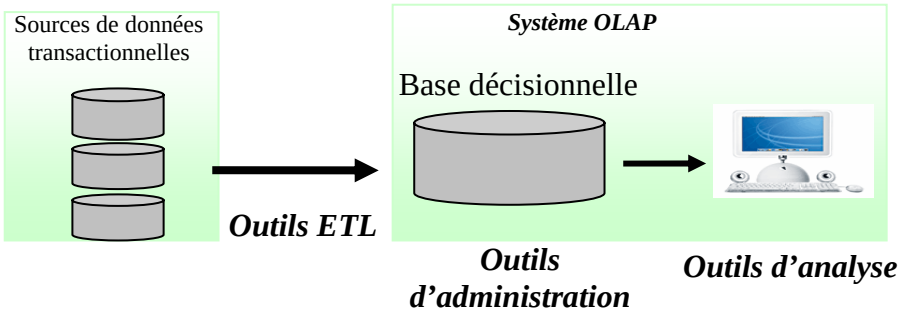


***Figure A.1** : Les outils industriels dans l'architecture du système décisionnel*

Les outils d'administration des bases de données permettent le stockage et l'administration des données de l'entrepôt de données. Ces outils sont disponibles pour assister l'administrateur dans la construction des entrepôts et des magasins de données *(Oracle OLAP*[1]*, SAS/Warehouse Administrator*[2]). La plupart de ces outils ne proposent pas des techniques d'archivages de données nécessaires pour résumer le grand volume de données dans l'entrepôt.

Les outils de constitution (ETL) permettent d'extraire les données des bases de production, de les transformer et de les charger (ETL : Extraction, Transformation, Loading). En général, les outils d'administration réalisent ces fonctionnalités ; nous retrouvons par exemple *Oracle OLAP, SAS/Warehouse Administrator et DTS* de *SQL Server*[3].

Les outils de restitution rassemblent l'ensemble des outils utilisés pour l'analyse dimensionnelle des données (*Powerplay*[4]*)*, le Reporting, les outils de requêtes (*Business Intelligence*[5]) et le DataMining. Ils permettent d'accéder aux données contenues dans un entrepôt ou dans un magasin et les transforment en informations exploitables par l'utilisateur. Ces outils ne proposent pas de démarche d'aide à la conception et de construction du système décisionnel puisqu'ils se basent sur les données de l'entrepôt ou des magasins déjà construits.

Nous présentons dans ce qui suit les principaux outils commerciaux dans le domaine d'administration, de constitution et de restitution de données.

Oracle Express et Analysis Server regroupent les fonctionnalités proposées par Oracle dans le domaine de l'analyse décisionnelle. *Oracle Express* est doté d'un SGBD multidimensionnelle. Il se base sur une approche MOLAP et permet de construire une base de données dimensionnelles à partir des bases transactionnelles. Analysis Server permet d'interroger et de manipuler ces données afin de fournir un outil d'analyse aux décideurs. Cette offre est caractérisée par l'extension du langage SQL pour supporter la gestion des données dimensionnelles. Ainsi, la commande ***Create Dimension*** est intégrée dans le langage de définition de données et la Clause ***group by*** du langage d'interrogation est étendu par les commandes ***Cube*** et ***RollUp***. Nous remarquons que cet outil nécessite une bonne expertise de l'administrateur de la base décisionnelle tout au long du processus de conception de celle-ci.

[1] *http://otn.oracle.com/products/oracle9i/olap.pdf*
[2] *http://www.sas.com/products/wadmin/*
[3] *.http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsql2k/html/dts_overview.asp*
[4] *www.**cognos**.com/products/businessintelligence/analysis/*
[5] *www.businessobjects.com/*

COGNOS PowerPlay 7, numéro un mondial des analyses OLAP, ce logiciel se présente sous la forme d'un tableur muni de fonctions évoluées. Les auteurs le décrivent comme un « Outil d'analyse et de reporting des performances d'activité pour les sources de données OLAP, en environnement Web, Windows ou Excel ». Le système permet l'analyse d'un cube, à savoir un fait unique en fonction de plusieurs dimensions (un schéma en étoile).

La représentation conceptuelle du cube se fait sous la forme de l'arborescence bien connue de tout explorateur de données. Par exemple, l'explorateur de fichiers du système Windows dispose d'une présentation de l'arborescence des répertoires dans sa partie gauche.

Les requêtes se font par glisser déplacer des éléments (mesures, dimensions, hiérarchies paramètres et attributs faibles) vers la fenêtre de présentation des résultats (un tableur), en les positionnant en lignes ou en colonnes.

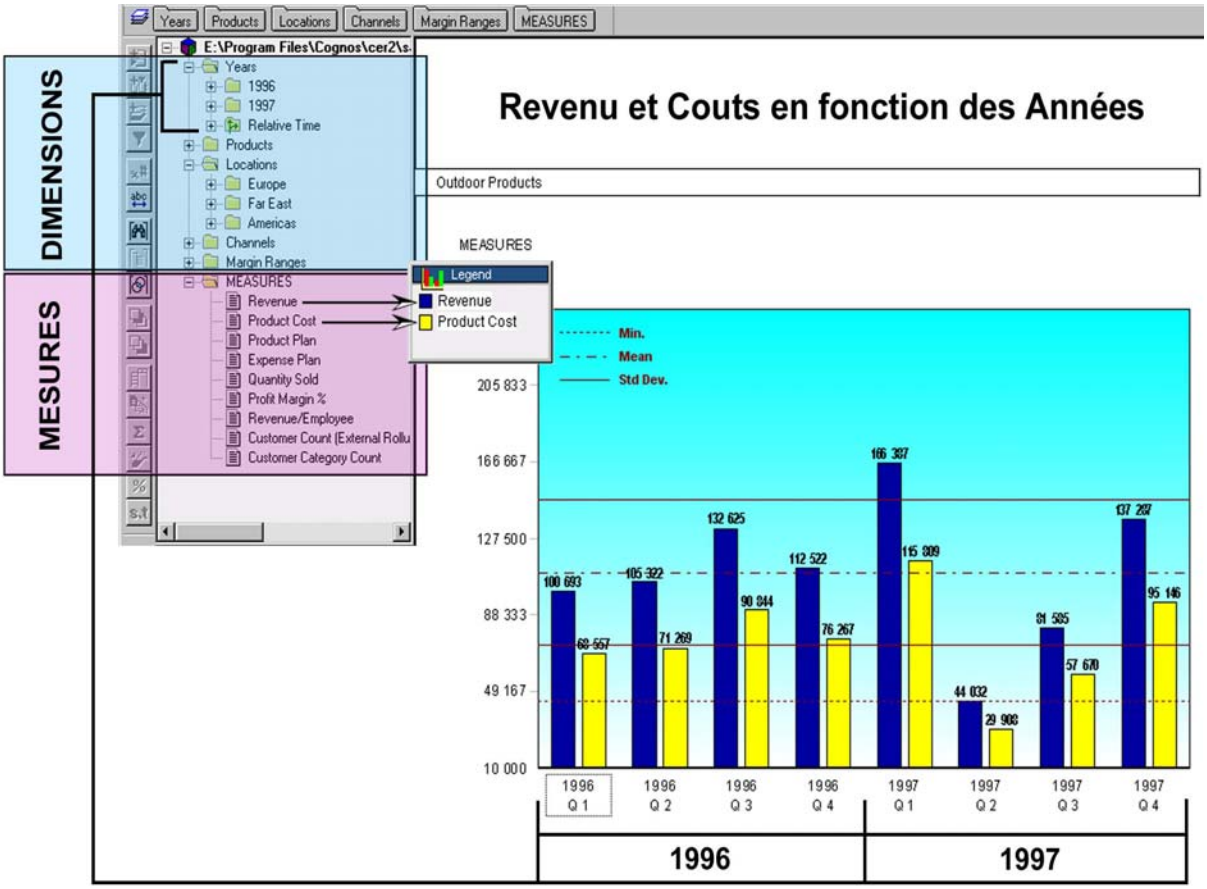


Figure A.2 *: PowerPlay, Analyse des Revenus et des Coûts en fonction des Années.*

CRYSTAL Analysis 10[6] est développé par Crystal Decisions, récente acquisition du géant Business Objects. CRYSTAL est un outil de visualisation de résultats et une extension pour Excel. Il propose pour la visualisation des données conceptuelle la même arborescence utilisée par tous les concurrents et se limite aussi à la visualisation d'un seul fait à la fois (une étoile / star). Des fonctionnalités avancées de sélection sont mises en avant mais elles requièrent toutes de maîtriser un langage de requête textuel (SQL en général).

[6] *www.**crystal**decisions.com/products/**crystalreports**/*

***Figure A.3** : Crystal Analysis, Captures d'écrans.*

Parmi les concurrents, on trouve aussi **Microstrategy** 7i[7] et **Business Objects** 6. Ces solutions logicielles fournissent les mêmes fonctionnalités en ce qui concerne la visualisation des données conceptuelles que ***PowerPlay*** et ***Crystal Analysis***. Ils ont comme avantage de pouvoir déployer leur propre serveur de données.

[7] *www.**microstrategy**.com/Software/*

TABLE DES FIGURES

Chapitre I. Contexte de l'étude

Chapitre II. Proposition d'un modèle dimensionnel contraint

Chapitre III. Interrogation des données dimensionnelles sous contraintes

Chapitre IV. Méthode de conception d'un schéma dimensionnel contraint

Chapitre V. Outil d'aide à la conception de magasin dimensionnel contraint

Chapitre VI. Annexe : outils industriels

LISTE DES TABLEAUX

Nom Prénom : Faiza GHOZZI JEDIDI

Directeur de Thèse : Gilles ZURFLUH

Lieu et date de soutenance : Université Toulouse III, IRIT, le 18 novembre 2004

Titre : **CONCEPTION ET MANIPULATION DE BASES DE DONNEES DIMENSIONNELLES A CONTRAINTES**

RESUME :
Dans le cadre des systèmes décisionnels, mes travaux de thèse consistent à étudier la modélisation des données dimensionnelles et à proposer un langage de manipulation adapté. Nous proposons un modèle dimensionnel organisant les données en une constellation de faits (sujets d'analyse) associés à des dimensions (axes d'analyse). Ces dimensions se caractérisent par leur flexibilité car chaque instance de la dimension peut appartenir à une ou plusieurs hiérarchies (perspectives d'analyse). L'intégration des contraintes dans le modèle dimensionnel a permis de valider la structure dimensionnelle et de désambiguiser les valeurs nulles provenant de la combinaison de hiérarchies incohérentes. Afin de tenir compte de ces contraintes, nous avons défini une algèbre d'interrogation des données dimensionnelles permettant au décideur de préciser les données à analyser. Enfin, nous proposons une méthode de conception d'un schéma dimensionnel intégrant les besoins des décideurs et les données sources. Pour valider nos travaux, nous avons développé un outil d'aide à la conception de schéma dimensionnel contraint.

Mots clés
Système décisionnel, modèle dimensionnel, contraintes, méthode de conception, algèbre d'interrogation.

TITLE : DESIGN AND QUERYING OF CONSTRAINT MULTIDIMENSIONAL DATABASES

ABSTRACT :
In decisional systems framework, my thesis focuses on the conception and the querying of multidimensional data. We provide a dimensional model organising data in a constellation of facts (subject of analysis) associated with dimensions (axes of analysis). These dimensions are flexible because each dimension instance can belong to one or several hierarchies (perspectives of analysis). Moreover, we integrate constraints in the multidimensional model allowing both the validation of multidimensional structures and the ensure of consistent analyses by disambiguating null values which can result from the combination of unsuitable hierarchies. To integrate these constraints, we define a multidimensional algebra enabling decision-makers to precise instances to analyse. Finally, we provide a conceptual design method of multidimensional data integrating decision-maker needs and data issued from source schema. To validate our propositions, we provide a CASE tool to help designer to define conceptual multidimensional schema.

Keywords:
Decisional system, Multidimensional model, Constraints, Design method, querying algebra

INSTITUT DE RECHERCHE EN INFORMATIQUE DE TOULOUSE
Centre National de la Recherche Scientifique (UMR 5505) - Institut National Polytechnique - Université Paul Sabatier – Université Toulouse 1
Université Paul Sabatier, 118 route de Narbonne, 31062 Toulouse cedex 04, Tel. 05.61.55.67.65